2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

BLOCK LANCZOS ALGORITHM

by

Yong Joo Kim

December 1989

Thesis Advisor: Murali Tummala

Approved for public release; distribution is unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 62
7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

BLOCK LANCZOS ALGORITHM

12 PERSONAL AUTHOR(S)

KIM, Yong Joo

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	1989 December	67

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	single vector; block Lanczos; DOA

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

We use a Lanczos algorithm for computing a few of the smallest eigenvalues and the corresponding eigenvectors of a large symmetric matrix rather than computing all the eigenvalue-eigenvector pairs. The basic Lanczos algorithm generates a similar matrix which is block tridiagonal from a given large symmetric matrix. The size of the generated tridiagonal matrix depends upon the number of the small eigenvalues to be computed. The result is the savings in computations and storage. The block Lanczos algorithm is well suited for problems involving multiple eigenvalues.

In this thesis, we develop the block Lanczos algorithm to estimate the direction-of-arrival (DOA) of a point source based on the observations measured at a linear array of sensors and compare the performance with that of a single vector Lanczos algorithm. The results of the computer simulation experiments conducted with this method is presented and discussed. Keywords: [illegible], Reorthogonalization. (KR)

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL TUMMALA, Murali	22b TELEPHONE (Include Area Code) 408-646-2645
22c OFFICE SYMBOL 62Tu	

DD Form 1473, JUN 86 — Previous editions are obsolete — SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603 — UNCLASSIFIED

Appproved for Public Release; Distribution is Unlimited

BLOCK LANCZOS ALGORITHM

by

Yong Joo Kim
Captain, Korean Army
B.S.E.E., Korea Military Academy, 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1989

Author: ______________________
Yong Joo Kim

Approved by: ______________________
Murali Tummala, Thesis Advisor

____
Charles W. Therrien, Second Reader

____
John P. Powers, Chairman,
Department of Electrical and Computer Engineering

ABSTRACT

We use a block Lanczos algorithm for computing a few of the smallest eigenvalues and the corresponding eigenvectors of a large symmetric matrix rather than computing all the eigenvalue-eigenvector pairs. The basic Lanczos algorithm generates a similar matrix which is block tridiagonal from a given large symmetric matrix. The size of the generated tridiagonal matrix depends upon the number of the smallest eigenvalues to be computed. The result is savings in computations and storage. The block Lanczos algorithm is well-suited for problems involving multiple eigenvalues.

In this thesis, we develop the block Lanczos algorithm to estimate the direction-of-arrival (DOA) of a point source based on the observations measured at a linear array of sensors and compare the performance with that of a single vector Lanczos algorithm. The results of the computer simulation experiments conducted with this method are presented and discussed.

TABLE OF CONTENTS

I. INTRODUCTION

It is necessary to estimate the power spectral density (PSD) function in several signal processing problems. Of particular interest is the estimation of spectral lines in noise. One related problem is the direction–of–arrival estimation of point sources based on array measurements. Methods based on subspace estimation have been proposed in this regard [Ref. 9]. Much of the basic literature on these methods has been borrowed from functional and numerical analysis [Refs. 8, 21]. Determination of the subspaces needs a complete eigendecomposition of the given data autocorrelation matrix. The general eigendecomposition of an n×n matrix requires $O(n^3)$ computations. Often we only need to compute a few of either the smallest or the largest eigenvalues and the corresponding eigenvectors of a large symmetric autocorrelation matrix rather than all the eigenpairs of the matrix. In this research, we develop an algorithm for computing a few of the smallest eigenvalues and the corresponding eigenvectors and apply this to high resolution spectral estimations problems.

The algorithm we develop is an extension of the method of minimized iterations due to Lanczos [Ref. 9]. Paige [Ref. 4] experimented with the Lanczos algorithm and found that a few of the extreme eigenvalues of a tridiagonal matrix would often converge rapidly to the similar eigenvalues of a real symmetric matrix R much before the entire set of eigenvalues are computed.

The block Lanczos algorithm is an extension of the basic single vector Lanczos algorithm. We iterate with a block of vectors rather than with a single vector, and generate a reduced block tridiagonal matrix that is similar to the original autocorrelation matrix R. The basic idea is that the eigenvalues of the block tridiagonal matrix are approximately the same as the extreme eigenvalues of R. Several researchers have reported the block Lanczos algorithm and its variants, in particular, Cullum and Willoughby [Ref. 1], Kahan and Parlett [Ref. 6] and Golub and Underwood [Ref. 4].

The objective of this thesis is to develop a block Lanczos algorithm to extract a few of the smallest eigenvalues and then estimate the corresponding eigenvectors. The smallest eigenvalues are said to correspond to the noise subspace of the spectral or array measurements. The proposed algorithm will be used to estimate the spectral lines which may represent the direction-of-arrival of point sources in low signal-to-noise ratio environments. The block Lanczos algorithm will have to be compared with the single vector case with respect to the spectral line estimation performance. With these goals in mind, we now proceed to discuss how the thesis is organized.

In Chapter II we introduce and summarize the single vector Lanczos algorithm. Complete and selective reorthogonalization of Lanczos vectors is discussed there. In Chapter III, we describe and develop the block Lanczos algorithm and present the results of the experiments carried out with a computer program implementing this algorithm. Also, in this chapter, we compare the performance of DOA estimation of the block Lanczos algorithm with that of the single vector Lanczos algorithm. Finally, in the last chapter, we discuss and summarize the results of the Lanczos method and also make some recommendations for future work.

II. SINGLE VECTOR LANCZOS ALGORITHM

The single vector Lanczos algorithm is used to tridiagonalize a real symmetric matrix R. This algorithm is based on the concepts such as Krylov sequences and subspaces, orthogonal projections of matrices, and Ritz vectors [Ref. 1]. In this chapter we describe the general Lanczos recursion and a practical Lanczos algorithm. Issues related to the reorthogonalization of Lanczos vectors are also discussed.

A. LANCZOS RECURSION

Given an n×n real symmetrix matrix R and an arbitrary n×1 unit norm vector $\mathbf{k}_1$, we can obtain a sequence of vectors defining the n dimensional Krylov subspace as follows [Ref. 1]

$$\begin{aligned}
\mathbf{k}_2 &= R\mathbf{k}_1 \\
\mathbf{k}_3 &= R\mathbf{k}_2 = R^2\mathbf{k}_1 \\
&\vdots \\
\mathbf{k}_n &= R\mathbf{k}_{n-1} = R^{n-1}\mathbf{k}_1.
\end{aligned} \tag{2–1}$$

We can now form the Krylov matrix of rank m as follows

$$\begin{aligned}
K_m &= [\mathbf{k}_1\ \ \mathbf{k}_2\ \ \mathbf{k}_3\ \cdots\ \mathbf{k}_m] \\
&= [\mathbf{k}_1, R\mathbf{k}_1, R^2\mathbf{k}_1, \cdots, R^{m-1}\mathbf{k}_1]
\end{aligned} \tag{2–2}$$

and the Krylov subspaces $\mathscr{K}^m$, m=1,2,$\cdots$,n, are then defined as

$$\mathscr{K}^m(R,\mathbf{k}_1) = \text{span}\,\{\mathbf{k}_1, R\mathbf{k}_1, R^2\mathbf{k}_1, \cdots, R^{m-1}\mathbf{k}_1\}. \qquad (2\text{–}3)$$

We now proceed to obtain a tridiagonal matrix which has the same extreme eigenvalues as the given symmetric matrix R. If the columns of a matrix $\mathbf{Q}_m$ are orthonormal, then the matrix

$$T_m = \mathbf{Q}_m^T R \mathbf{Q}_m \qquad (2\text{–}4)$$

is an m×m tridiagonal matrix which is an orthogonal projection of R onto the space spanned by the columns of $\mathbf{Q}_m$. Lanczos [Refs. 1, 9, 14] proposed a recursion to generate the columns of the matrix $\mathbf{Q}_m = [\mathbf{q}_1\ \mathbf{q}_2 \cdots \mathbf{q}_m]$. The vectors $q_1, q_2, \cdots, q_m$ are referred to as Lanczos vectors. Thus, for a given real symmetric matrix R, the Lanczos recursion produces a tridiagonal matrix T_m as a projection of R onto the corresponding subspace span$\{\mathbf{Q}_m\}$, spanned by the Lanczos vectors generated. These subspaces are in fact Krylov subspaces. [Ref. 1]

The classical Lanczos recursion for R starts with an n×1 initial Lanczos vector $\mathbf{q}_1$ which is randomly generated and normalized. Initially we define $\beta_1 \equiv 0$ and $\mathbf{q}_0 \equiv \mathbf{0}$. We then compute α_i, β_{i+1} and the Lanczos vectors $\mathbf{q}_i$ for i=1,2,$\cdots$,m as follows:

$$\alpha_i = \mathbf{q}_i^T R \mathbf{q}_i \qquad (2\text{–}5)$$

$$\beta_{i+1}\mathbf{q}_{i+1} = R\mathbf{q}_i - \alpha_i\mathbf{q}_i - \beta_i\mathbf{q}_{i-1} \qquad (2\text{–}6)$$

$$\beta_{i+1} = \mathbf{q}_{i+1}^T \mathbf{R}\mathbf{q}_i. \tag{2–7}$$

The Lanczos matrix T_m is generated by appropriately arranging α_i and β_{i+1}, where α_i are the diagonal elements and β_{i+1} are the subdiagonal elements of the tridiagonal matrix, so that

$$T_m(i,i) = \alpha_i \qquad \text{for } 1 \leq i \leq m, \tag{2–8}$$

and

$$T_m(i,i+1) = T_m(i+1,i) = \beta_{i+1} \qquad \text{for } 1 \leq i \leq m-1, \tag{2–9}$$

resulting in

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \beta_m \\ & & & & & \beta_m & \alpha_m \end{bmatrix}. \tag{2–10}$$

The vectors $\alpha_i\mathbf{q}_i$ and $\beta_i\mathbf{q}_{i-1}$ in Eqn(2–6) are the orthogonal projections of vectors $\mathbf{R}\mathbf{q}_i$ onto the two most recently generated Lanczos vectors $\mathbf{q}_i$ and $\mathbf{q}_{i-1}$. Thus, updated Lanczos vector $\mathbf{q}_{i+1}$ is computed by orthogonalizing the vector $\mathbf{R}\mathbf{q}_i$ with respect to previously computed Lanczos vectors $\mathbf{q}_i$ and $\mathbf{q}_{i-1}$. The classical Lanczos recursion can be condensed into matrix notation by a group of Lanczos vectors $\mathbf{Q}_m$

and the Lanczos matrix T_m. We obtain the following matrix equation

$$RQ_m = Q_m T_m + \beta_{i+1} q_{i+1} e_i^T \tag{2–11}$$

where e_i is the unit vector whose i^{th} element is 1 and whose other elements are 0. Note that, in this recursion, R is never modified. Also, storage is needed only for the Lanczos vectors q_{i-1}, q_i and q_{i+1}, the Lanczos matrix T_m, and the given matrix R.

The classical Lanczos procedure attempts to maintain the orthogonality of the Lanczos vectors. Due to the roundoff and other numerical errors, however, orthogonality between the Lanczos vectors can only be maintained by incorporating some kind of explicit reorthogonalization as the Lanczos vectors are computed. Note that the reorthogonalization of these vectors requires extra storage for keeping all of the Lanczos vectors as well as additional computations. We discuss the reorthogonalization in a later section.

B. PRACTICAL LANCZOS ALGORITHM

In this section we focus on a Lanczos algorithm with no explicit reorthogonalization incorporated. The loss of orthogonality, if it occurs, may result in spurious eigenvalues. Although the orthogonality of the Lanczos vectors is lost, some of the eigenvalues of R will still appear as eigenvalues of the Lanczos matrix if we make the tridiagonal matrix large enough. Since we are generally interested in only a few of the smallest eigenvalues and their corresponding eigenvectors, the loss of orthogonality does not critically affect finding approximate eigenvalues and corresponding eigenvectors of R [Ref. 1]. Nevertheless, it is possible to find accurate eigenvalues and eigenvectors by using the method in an iterative scheme using no reorthogonalization at all, even in the face of total loss of orthogonality [Ref. 12]. Now, we present the practical single vector Lanczos algorithm.

To generate the tridiagonal Lanczos matrix T_m we should compute α_i and β_{i+1} which are the diagonal and sub-diagonal elements of the tridiagonal matrix, where $i=1,2,\cdots,m$. Given an $n\times n$ real symmetric matrix R, we start with an $n\times 1$ initial arbitrary vector $\mathbf{q}_1$ which is normalized such that $\|\mathbf{q}_1\|_2=1$ as in Eqn(2–1). We now define an intermediate vector

$$\mathbf{u}_1 = \mathrm{R}\mathbf{q}_1 \tag{2–12}$$

and initialize $\alpha_0=0$ and $\beta_0=0$. The single vector recursion is then carried out for $i=1,2,\cdots,m$ and can be summarized as follows:

$$\alpha_i = \mathbf{q}_i^T\mathbf{u}_i \tag{2–13}$$

$$\mathbf{w}_i = \mathbf{u}_i - \alpha_i\mathbf{q}_i \tag{2–14}$$

$$r_{i+1} = +\sqrt{\mathbf{w}_i^T\mathbf{w}_i} \tag{2–15}$$

$$\mathbf{q}_{i+1} = \mathbf{w}_i/r_{i+1} \tag{2–16}$$

$$\beta_i = \mathbf{q}_i^T\mathrm{R}\mathbf{q}_{i+1} \tag{2–17}$$

$$\mathbf{u}_{i+1} = \mathrm{R}\mathbf{q}_{i+1} - \beta_i\mathbf{q}_i \tag{2–18}$$

where $\mathbf{u}_i$ and $\mathbf{w}_i$ are the orthonormal projection vectors for the Lanczos vector $\mathbf{q}_i$. Here Eqn(2–13) and Eqn(2–17) use the modified Gram-Schmidt orthogonalization to compute the coefficients α_i and β_{i+1}, respectively. If $\mathbf{q}_i$ is orthogonal to $\mathbf{q}_{i-1}$, then $\mathbf{q}_{i+1}$ will theoretically be orthogonal to both q_{i-1} and q_i. This algorithm is quite popular despite the requirement for small amounts of extra computations [Ref. 12]. Now the tridiagonal matrix T_m is generated by simply filling it with α_i and β_i for its entries as shown in Eqn(2–10).

1. Eigenvalue Computation

To find the eigenvalues μ_i of the m×m Lanczos matrix T_m, we may use the bisection method and Sturm sequencing [Ref. 5]. Actually, one could obtain both eigenvalues and eigenvectors of T_m by using such methods as the QR algorithm [Ref. 16]. However, since we need only a few of the smallest eigenvalues of T_m, we choose the bisection method to compute them as detailed in the following.

Given the tridiagonal matrix T_m, we define the characteristic polynomials $p_0(\mu)$, $p_1(\mu)$, $\cdots$, $p_m(\mu)$ as

$$p_0(\mu) = 1 \tag{2–19}$$

$$p_j(\mu) = \det(T_m - \mu I), \tag{2–20}$$

for j=1,2,$\cdots$,m. Expanding the determinant in Eqn(2–20) yields the recursive expression [Ref. 5: pp. 305–307]:

$$p_j(\mu) = (\alpha_j - \mu)p_{j-1}(\mu) - \beta_j^2 p_{j-2}(\mu), \qquad j=2,3,\cdots,m. \tag{2–21}$$

The zeros of the polynomial $p_m(\mu)$ are the eigenvalues of the tridiagonal matrix T_m. Here we are interested in only a few of the smallest eigenvalues of R. In order to compute these values we first define a range [x, y] in which all the desired eigenvalues lie. We then carry out the following iteration to implement the Sturm sequencing property. If $p(x)p(y) < 0$ and $x < y$, then the iteration

$$|x - y| > \epsilon(\,|x| + |y|\,)$$

$$\mu = \frac{x + y}{2}$$

$$y = \mu \quad \text{if} \quad p_m(x)p_m(y) < 0 \tag{2–22}$$

$$x = \mu \quad \text{if} \quad p_m(x)p_m(y) \geq 0$$

is guaranteed to converge to a zero of $p_m(\mu)$, *i.e.*, to an eigenvalue of T_m. The value ϵ is the machine unit roundoff error and the limits on the range [x, y] are given by

$$\begin{aligned} x &= \min_i \alpha_i - |\beta_i| - |\beta_{i-1}| \\ y &= \max_i \alpha_i + |\beta_i| + |\beta_{i-1}| \end{aligned} \tag{2–23}$$

where we have $\beta_0 = \beta_{m+1} = 0$. The bisection method computes the eigenvalues with small relative error, regardless of their magnitude. This is in contrast to the tridiagonal QR iteration, where the computed eigenvalues can have only small absolute error.

2. Eigenvector Computation

Now, we need to compute the eigenvectors of R to complete the eigenpair estimation problem. There are two techniques to compute the approximate eigenvectors of R. The first technique uses the Ritz vector $\mathbf{x}_j$ and the corresponding algorithm uses the relationship [Ref. 14]

$$\mathbf{x}_j = \mathbf{Q}_m \mathbf{z}_j, \qquad j=1,2,\cdots,s \tag{2–24}$$

where $\mathbf{z}_j$ is one of the eigenvectors of T_m, $\mathbf{Q}_m$ is the matrix of Lanczos vectors, and $s \leq \frac{m}{2}$.

The second technique to obtain the eigenvectors of R corresponding to the selected eigenvalues of T_m uses the Rayleigh quotient iteration. The iteration makes the Rayleigh quotient $r(\mathbf{x}_k)$ converge to μ_i which is one of the smallest eigenvalues of T_m. The Rayleigh quotient of an eigenvector $\mathbf{x}_k$ which minimizes $\|(R - r_k I)\mathbf{x}_k\|_2$ is given by

$$r_k = r(\mathbf{x}_k) = \frac{\mathbf{x}_k^T R \mathbf{x}_k}{\mathbf{x}_k^T \mathbf{x}_k}. \tag{2–25}$$

As $r(\mathbf{x}_k)$ approaches an eigenvalue μ_i of T_m, then the solution to $(R - r_k I)\mathbf{x}_k = \mathbf{b}_k$ will be an approximate eigenvector by using the inverse iteration theory where $\mathbf{b}_k$ is a vector close to zero [Ref. 14].

We now briefly present the inverse iteration algorithm. First, we pick an arbitrary unit vector $\mathbf{x}_0$; then the iteration proceeds as follows for $k = 0,1,\cdots\cdot\cdot$

$$r_k^{(i)} = \mathbf{x}_k^{(i)T} R \mathbf{x}_k^{(i)}, \qquad \mathbf{x}_k^{(i)T}\mathbf{x}_k^{(i)} = 1$$

$$\text{solve } (R - r_k^{(i)} I)\mathbf{y}_{k+1} = \mathbf{x}_k^{(i)} \text{ for } \mathbf{y}_{k+1} \tag{2-26}$$

$$\mathbf{x}_{k+1}^{(i)} = \mathbf{y}_{k+1}/\|\mathbf{y}_{k+1}\|_2.$$

If $r_k^{(i)}$ converges to one of the smallest eigenvalues μ_i of T_m where $i=1,2,\cdots,s$, then we stop the iteration and the corresponding vector $\mathbf{x}_{k+1}^{(i)}$ is the required eigenvector of R.

Having discussed the Lanczos algorithm and the methods to compute the desired eigenpairs, we now proceed to address some issues related to the reorthogonalization of the Lanczos vectors.

C. REORTHOGONALIZATION

As mentioned in the previous sections, the Lanczos vectors $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_m$ lose mutual orthogonality as the number of steps m in the algorithm increases. The requirement that $\mathbf{Q}_m^T\mathbf{Q}_m = I_m$ is then destroyed by the roundoff errors and the algorithm is described as unstable. A few steps later, the matrix of Lanczos vectors $\mathbf{Q}_m$ may not even have full rank, *i.e.*, the Lanczos vectors may become linearly dependent. As a result there is no guarantee that T_m will bear any useful relationship to R. And the orthogonality among vectors $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_m$ disappears. The ideal Lanczos algorithm should terminate ($\beta_m = 0$) for some $m \leq n$, but in practice the process goes on forever computing more and more spurious eigenvectors for each correct eigenpair it discovers. [Ref. 14]

Thus, sometimes we need to maintain the orthogonality of the Lanczos vectors for finding more accurate eigenvalues and eigenvectors of R and to avoid the computation of any spurious pairs. We now add the reorthogonalization step in the single vector Lanczos procedure presented in Section A and B of this Chater. There are two techiques for reorthogonalization, namely, complete reorthogonalization and selective reorthogonalization. These are discussed in the following.

1. Complete Reorthogonalization

Complete reorthogonalization incorporates a Householder matrix computation into the Lanczos algorithm for producing Lanczos vectors that are orthogonal to within the working accuracy. This is effective at maintaining the stability of the system. The following step is inserted into the Lanczos algorithm (Eqns(2–13)–(2–18)) after computing a projection vector $\mathbf{w}_i$:

$$\mathbf{w}_i = \mathbf{w}_i - \mathbf{q}_j(\mathbf{q}_j^T\mathbf{w}_i), \text{ for } j=i,i-1,\cdots,2,1 \tag{2–27}$$

thus $\mathbf{w}_i$ is explicity orthogonalized against $\mathbf{q}_i$ and $\mathbf{q}_{i-1}$. If a Householder matrix P_i is determined so that $[P_0\ P_1\ \cdots\ P_i]^T[\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_i]$ is upper triangular, then it follows that the $(i+1)^{st}$ column of $[P_0\ \cdots\ P_i]$ is the desired unit vector. An example of a complete reorthogonalization Lanczos scheme is summarized below. First, we determine the initial Householder matrix $P_0 = I - \mathbf{v}_0\mathbf{v}_0^T/\mathbf{v}_0^T\mathbf{v}_0$ so that $P_0\mathbf{w}_0 = \mathbf{e}_1$. From Eqn(2–5) we can then obtain $\alpha_1 = \mathbf{q}_1^T R\mathbf{q}_1$ and implement the following recursion for $i=1,2,\cdots,m-1$: [Ref. 5: pp. 334–345]

$$\mathbf{r}_i = (R - \alpha_i I)\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1} \quad (\beta_0\mathbf{q}_0=0)$$

$$\mathbf{w} = (P_{i-1} \cdots P_0)\mathbf{r}_i \tag{2–28}$$

$$\text{determine } P_i = I - 2\mathbf{v}_i\mathbf{v}_i^T/\mathbf{v}_i^T\mathbf{v}_i$$

$$P_i\mathbf{w}_i = (\mathbf{w}_1, \cdots, \mathbf{w}_i, \beta_i, 0, \cdots, 0)^T$$

$$\mathbf{q}_{i+1} = (P_0 \cdots P_i)\mathbf{e}_{i+1}$$

$$\alpha_{i+1} = \mathbf{q}_{i+1}^T R\mathbf{q}_{i+1}.$$

In Eqn(2–28) the computed Lanczos vectors $\mathbf{q}_i$ are now mutually orthogonal to the working precision which follows from the roundoff properties of Householder matrices [Ref. 5]. However, the complete reorthogonalization of Lanczos vectors requires extra computations and all the computed vectors $\mathbf{q}_1\ \mathbf{q}_2 \cdots \mathbf{q}_i$, $\mathbf{w}_1\ \mathbf{w}_2 \cdots \mathbf{w}_i$ need to be stored. It negates any advantage of the Lanczos algorithm.

2. Selective Reorthogonalization

Selective reorthogonalization is computationally more efficient. A small modification to the exact version of the Lanczos algorithm ensures that the Lanczos vectors $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3 \cdots$ maintain the orthogonality. At the i^{th} step the Lanczos algorithm (Eqns(2–13) – (2–18)) produces the matrix of Lanczos vectors $\mathbf{Q}_i$, the tridiagonal matrix $T_i = \mathbf{Q}_i^T R\mathbf{Q}_i$, and the residual vector $\mathbf{w}_i = (R\mathbf{Q}_i - \mathbf{Q}_i T_i)\mathbf{e}_i$. The equation to compute the residual vectors $R\mathbf{Q}_i = \mathbf{Q}_i T_i + \mathbf{w}_i\mathbf{e}_i^T$ can then be rewritten as

$$\mathbf{R}\mathbf{Q}_i = \mathbf{Q}_i\mathrm{T}_i + \mathbf{w}_i\mathbf{e}_i^T - \mathrm{F}_i, \tag{2–29}$$

where F_i accounts for the roundoff errors. When the Lanczos vectors lose their orthogonality we have

$$\| \mathrm{I} - \mathbf{Q}_i^T\mathbf{Q}_i \| \leq k_i \tag{2–30}$$

where it is required to keep $k_i \leq k$ for some k in the interval $(n\epsilon, 0.01)$ and ϵ is the relative precision of the arithmetic.

The loss of orthogonality goes hand in hand with the convergence of a Ritz pair. Suppose that the symmetric QR algorithm [Ref. 16] is applied to T_i and renders the computed Ritz values $\theta_1, \cdots, \theta_i$ and a nearly orthogonal matrix of eigenvectors $\mathrm{Z}_i=[z_1,z_2,\cdots,z_i]$ of T_i. Then the Ritz vectors which are the approximate eigenvectors of R are given by

$$\mathbf{X}_i = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i] = \mathbf{Q}_i\mathbf{Z}_i. \tag{2–31}$$

We can show that the absolute value of the inner product between $\mathbf{q}_{i+1}$ and $\mathbf{x}_j$ is

$$|\mathbf{q}_{i+1}^T\mathbf{x}_j| \cong \frac{\epsilon \, \|\mathrm{R}\|_2}{|\beta_i|\,|\mathbf{z}_{ij}|} \tag{2–32}$$

where the denominator term can be approximated as

$$|\beta_j|\,|z_{ij}| \cong \|\mathrm{R}\mathbf{x}_j - \theta_j\mathbf{x}_j\|_2 \tag{2–33}$$

for j=1, 2, $\cdots$, i. The recently updated Lanczos vector $\mathbf{q}_{i+1}$ is forced to have a unwanted component in the direction of any converged Ritz vector. Consequently, instead of orthogonalizing $\mathbf{q}_{i+1}$ against all of the previous Lanczos vectors, we can achieve the same effect by orthogonalizing it against the converged Ritz vectors [Ref. 5]. A selective reorthogonalization method based on this technique is discussed in Parlett and Scott [Ref. 15]. In this method, a computed Ritz pair $(\theta,\mathbf{x})$ is considered a good approximation if it satisfies

$$\|R\mathbf{x} - \theta\mathbf{x}\|_2 \cong \sqrt{\epsilon}\|R\|_2 \tag{2–34}$$

where ϵ is the machine precision constant. After computing $\mathbf{q}_{i+1}$, it is orthogonalized against each good Ritz vector [Ref. 5]. Selective reorthogonalization prevents the computation of many spurious eigenvectors. This means that the extra computations and storage required in this method are less than those required to implement the complete reorthogonalization since there are usually fewer Ritz vectors than Lanczos vectors. When all of the eigenvalues of R are required, however, then the selective reorthogonalization procedures are too expensive to implement [Ref. 12].

D. RESULTS

In this section we will discuss the experimental results based on computer simulation of the single vector Lanczos algorithm for estimating the direction-of-arrival of point sources.

1. Experimental Set Up

Consider that we receive signals at a linear array containing M equally-spaced sensors. The signal is modeled as a sum of l sinusoids, the normalizing spatial frequencies of which are propotional to their bearings from 0.0 (= 0°) to 0.5 (= 90°), in additive random noise with fixed variance. The received signal at the m^{th} sensor location is then given by

$$z(m) = \sum_{i=1}^{l} A_i \cos(2\pi f_i m) + n(m) \tag{2–35}$$

where A_i is the amplitude of the i^{th} sinusoid, f_i is the normalized spatial frequency of the i^{th} signal which represents the bearing, and n(m) is zero-mean white Gaussian noise with variance σ^2. The amplitude of the signal A_i and the noise variance σ^2 will determine the signal to noise ratio of the i^{th} signal,

$$\mathrm{SNR}_i = 10 \log\left(\frac{\frac{A_i^2}{2}}{\sigma^2} \right). \tag{2–36}$$

Figure 1 shows the block diagram of the direction-of-arrival estimation algorithm considered in this work. After measuring the signals received at each sensor, the sensor output is passed through a bank of bandpass filters. This helps prefilter the noise over the selected frequency band of frequencies and provides some initial processing gain [Ref. 22]. The autocorrelation matrix is then computed by taking data from the outputs of corresponding bandpass filters at each sensor. An n×n autocorrelation matrix is generated as follows

$$R_{zz}(k) = \frac{1}{M} \sum_{m=0}^{M-1-k} z(m+k)z(m), \text{ for } 0 \le k \le n-1. \qquad (2\text{–}37)$$

We now have computed an n×n autocorrelation matrix of z(m), R_{zz}, by using M data samples. The eigenvectors $\mathbf{x}_i$ of R_{zz} corresponding to the smallest eigenvalues are computed by using the single vector Lanczos algorithm. The power spectral density estimates are computed as:

$$S_{xx}^{(i)}(f) = \left| \frac{1}{\sum_{j=1}^{n-1} x_{ij} z^{-j}} \right|^2_{z=e^{j2\pi f}} \qquad (2\text{–}38)$$

where x_{ij} are the elements of the i^{th} eigenvector x_i.

2. DOA Estimation

A few of the extreme eigenvalues and their corresponding eigenvectors of a large real symmetric matrix R can be obtained by using the single vector Lanczos algorithm. Those eigenpairs are the approximate eigenpairs of R. Each eigenvector has spectral information to determine the bearing of a source relative to an array of sensors. The physical implementation of a DOA estimation scheme is shown in Figure 1. The correlator generates an autocorrelation matrix of the received signal. The Lanczos algorithm and the eigendecomposition produce the noise subspace eigenvectors that estimate the spatial power spectral density (PSD) function, given in Eqn(2–38). The PSD function represents the direction-of-arrival of point sources as spectral peaks. In this thesis we have used three different ways of estimating the PSD function: the individual eigenvector spectra, the algebraic averaging of a few eigenvectors, and the spectral multiplication of the individual spectra.

The computer simulation experiment consists of an equally-spaced array of 100 sensors arranged in a linear fashion. By using a filter bank as shown in Figure 1, the signals have known temporal frequency with unknown bearings. The autocorrelation matrix size is chosen to be 25×25. The number of eigenpairs computed is chosen according to the number of iterations in the Lanczos algorithm. We have used 5 dB, 0 dB, –5 dB and –10 dB SNR cases for the direction-of-arrival estimation. The results indicate the ability of the algorithm to determine the number of targets and bearing resolution for various directions and different SNRs. In each case we have used the five smallest eigenvalues and their corresponding eigenvectors for computing the PSD function. (We can, however, choose more eigenvectors at the expense of more computations.)

We can use eigenvector averaging, which is the algebraic averaging of the computed eigenvectors corresponding to the smallest eigenvalues of R. It has improved the performance compared to that of the individual eigenvector spectra. Further improvement in results, however, can be obtained by spectral multiplication of the individual eigenvector spectra. The estimate is given by

$$S_{xx}(f) = \prod_{i=1}^{J} S_{xx}^{(i)}(f) \tag{2–39}$$

where J is a predetermined number ($J \leq m \leq n$).

We now consider several examples to study the estimation performance of the Lanczos algorithm and the consequent eigenpair computation. Most of the results use the Ritz vector method with no reorthogonalization. Example 1 is the detection of a single target at 9° for different SNRs. Figure 2 shows the overlayed individual spectra of the five eigenvectors corresponding to the smallest eigenvalues at an SNR of 5 dB. Note that the spectrum of each eigenvector has several spurious peaks, but each eigenvector has a common peak at the true bearing 9°. Figure 3(a) shows a plot of the PSD function which is the average of 5 eigenvectors. The averaging improves the estimation performance considerably. However, further improvement was obtained by using the spectral multiplication as defined in Eqn(2–39) and the result is shown in Figure 3(b). As can be seen, even though it has two small spurious peaks, it has greatly improved the nulls and the peak at 9°. In the remainder of the thesis, we use the spectral multiplication method to compute the PSD function. Figure 4 and 5 show the results at 0 dB and –5 dB SNR respectively. More spurious peaks are observed as the SNR is lowered. At an SNR of –10 dB (see Figure 6(a)), several spurious peaks are seen which are almost as large as the true bearing. Improved performance is obtained as shown in Figure 6(b) by using more eigenvectors (7 eigenvectors) in this case in the spectral multiplication.

In Example 2 we have 3 targets at 34°, 36° and 54°. Notice that two targets are very closly spaced in bearing. Figure 7(a) is obtained by using the Ritz vector method and Figure 7(b) is obtained by using the Rayleigh quotient iteration to compute the eigenvectors at 0 dB. Although the two targets are very closely spaced, good resolution is clearly achieved and almost no spurious peaks are seen in both results. Figure 8(a) and Figure 8(b) are obtained by using no reorthogonalization and with complete reorthogonalization respectively at an SNR of –5 dB. As mentioned earlier, loss of orthogonality does not affect the results for a few eigenvectors in the single vector Lanczos procedure. The result of reorthogonalization is almost the same as the result of no reorthogonalization. Resolution is achieved in both cases at –5 dB, but several spurious peaks are present. Figure 9(a) shows the result using 3 eigenvectors while Figure 9(b) indicates the performance using 5 eigenvectors at an SNR of –10 dB. Note that sufficient spectral resolution is not acheived to discriminate the targets located at 34° and 36°. A number of spurious peaks are higher than the peak at 36° making it impossible to accurately determine the number of targets as well as their locations.

The results in this chapter indicate that the eigenvectors found using the single vector Lanczos algorithm are sufficiently accurate to determine the spectrum. The spectral multiplication scheme achieves the best spectral estimation performance. This algorithm provides savings in computations and storage because it needs to compute only a few of the extreme eigenvalues and eigenvectors of a large real symmetric matrix. Loss of orthogonality is not critically affected when we need to find only a few eigenvalues and eigenvectors of a large autocorrelation matrix.

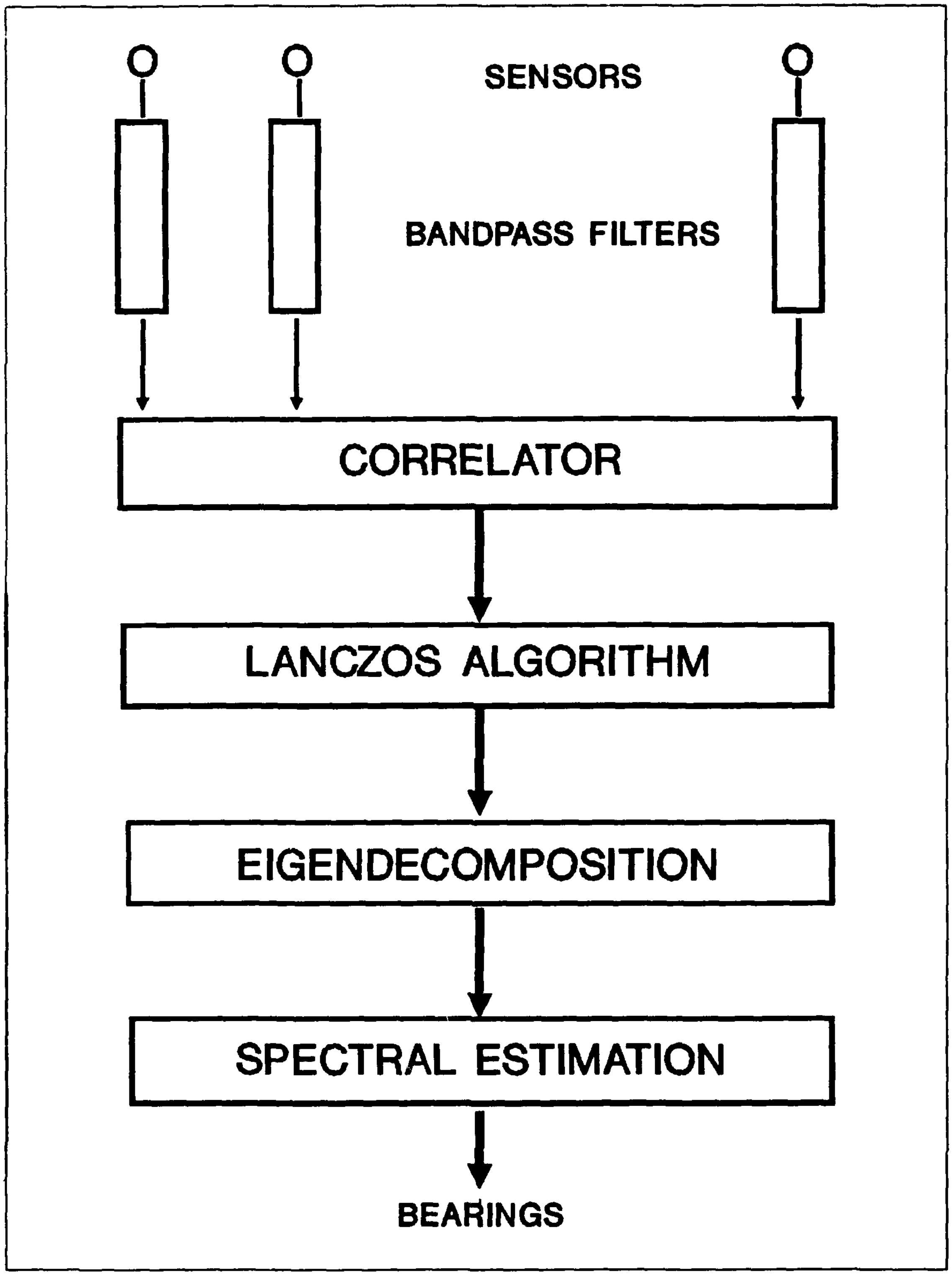

Figure 1. Experiment Set up

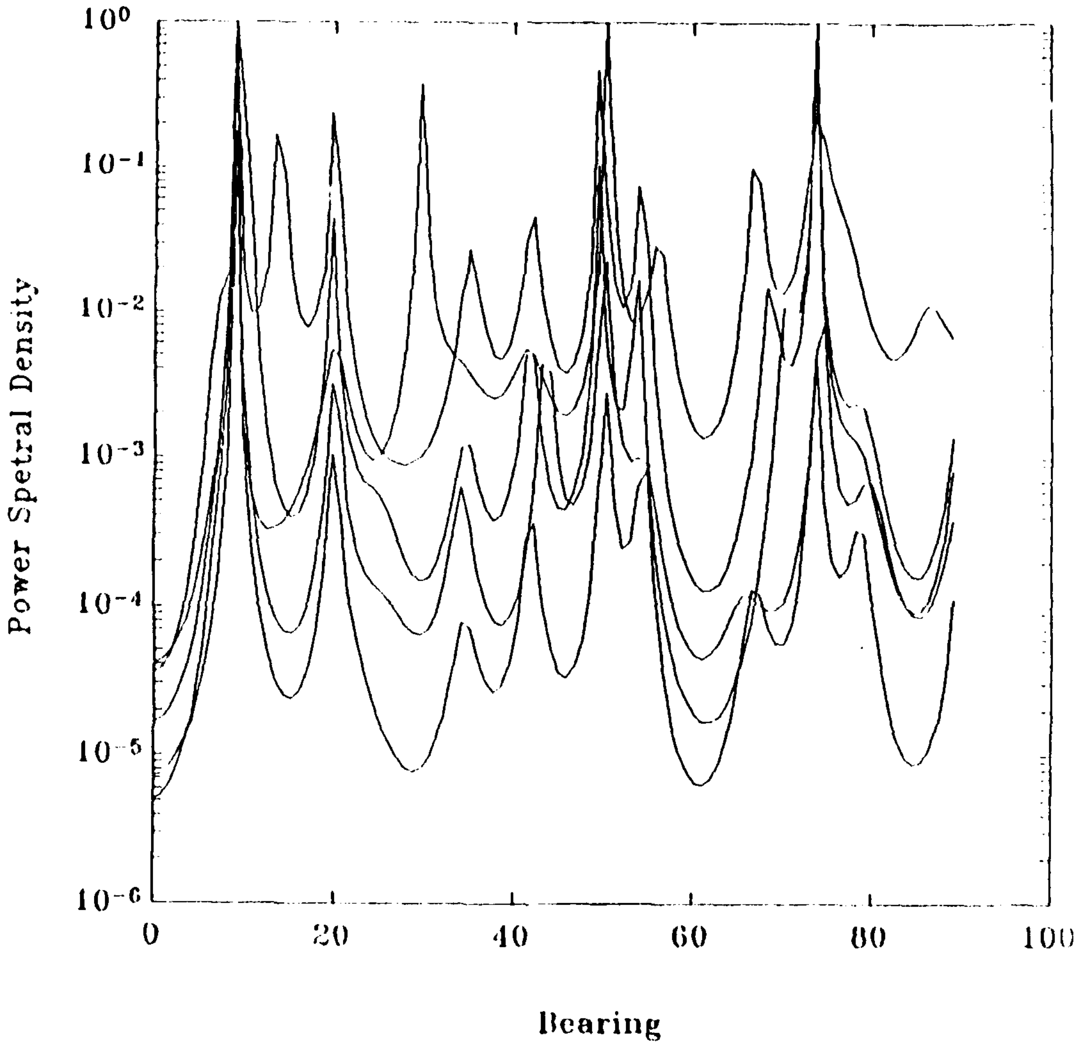

Figure 2. One target at 9° (5 dB) : overlay of 5 eigenvectors

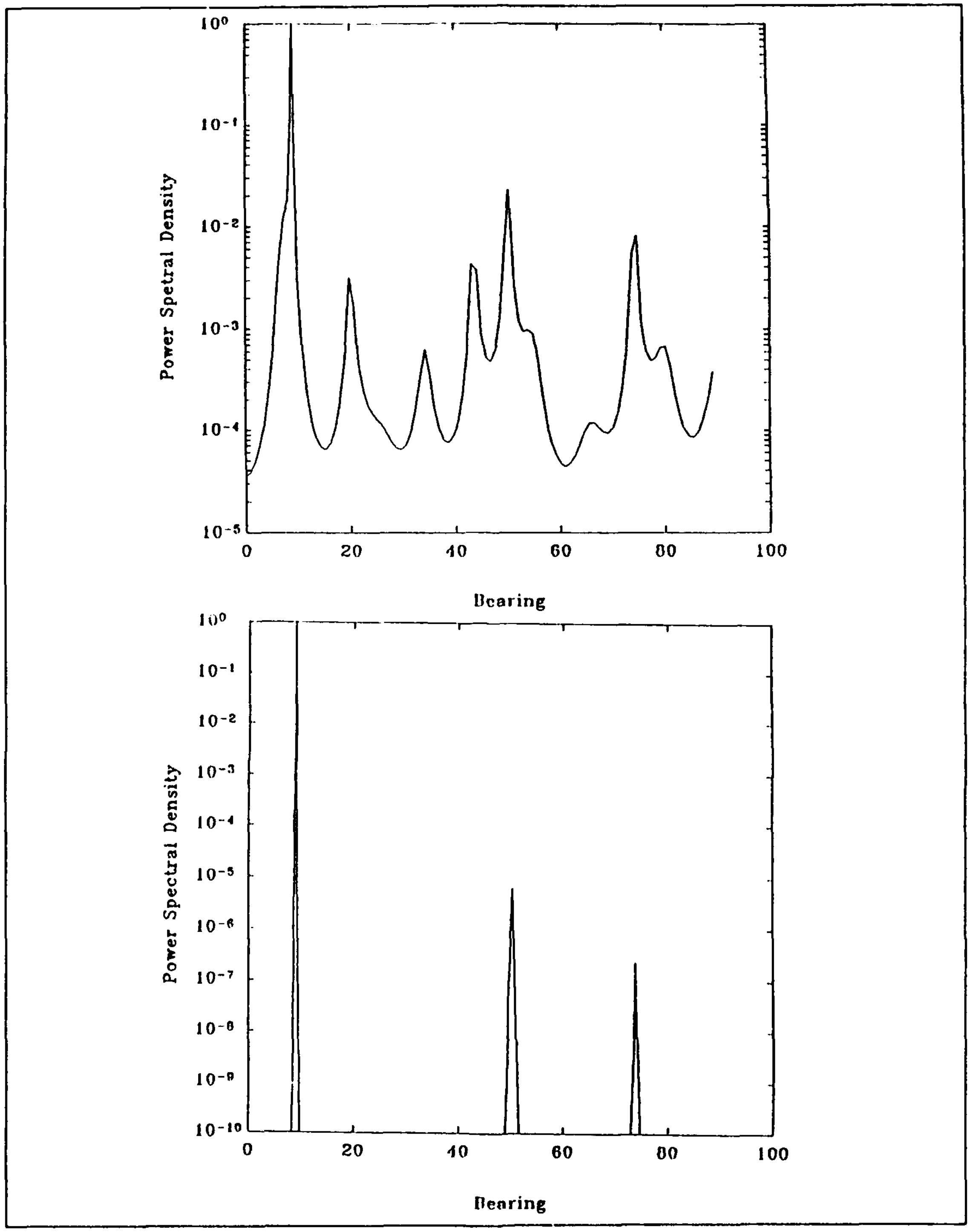

Figure 3. One target at 9° (5 dB) : (a) average of 5 eigenvectors
(b) product of the PSDs 5 eigenvectors

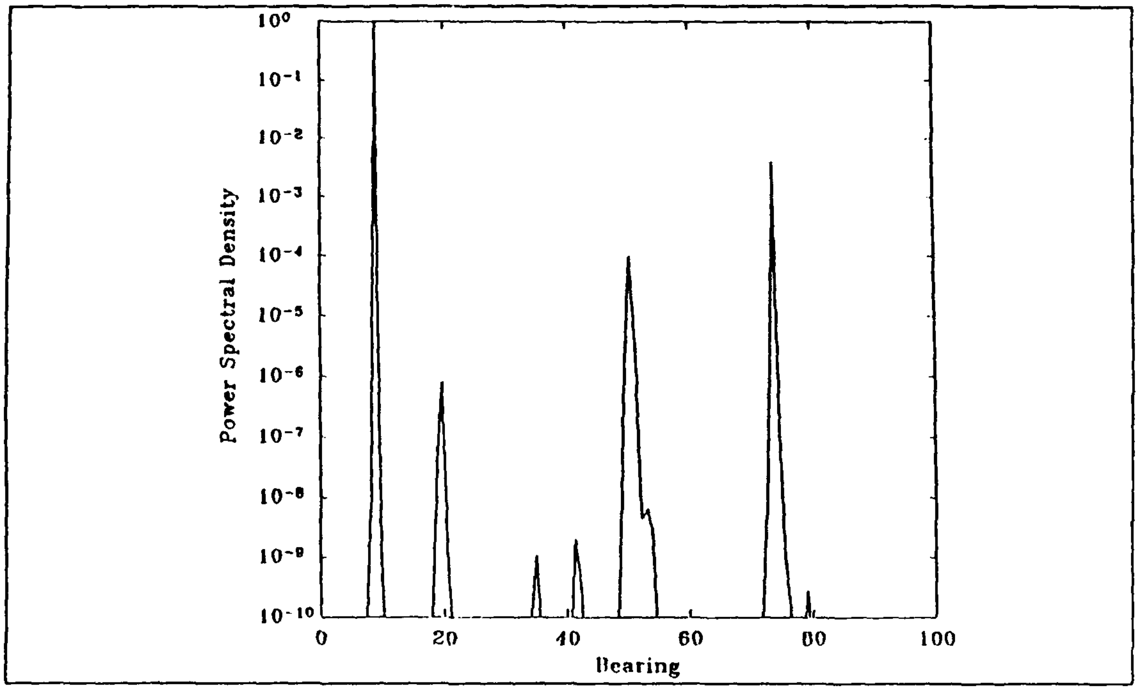

Figure 4. One target at 9° (0dB) : product of the PSDs 5 eigenvectors

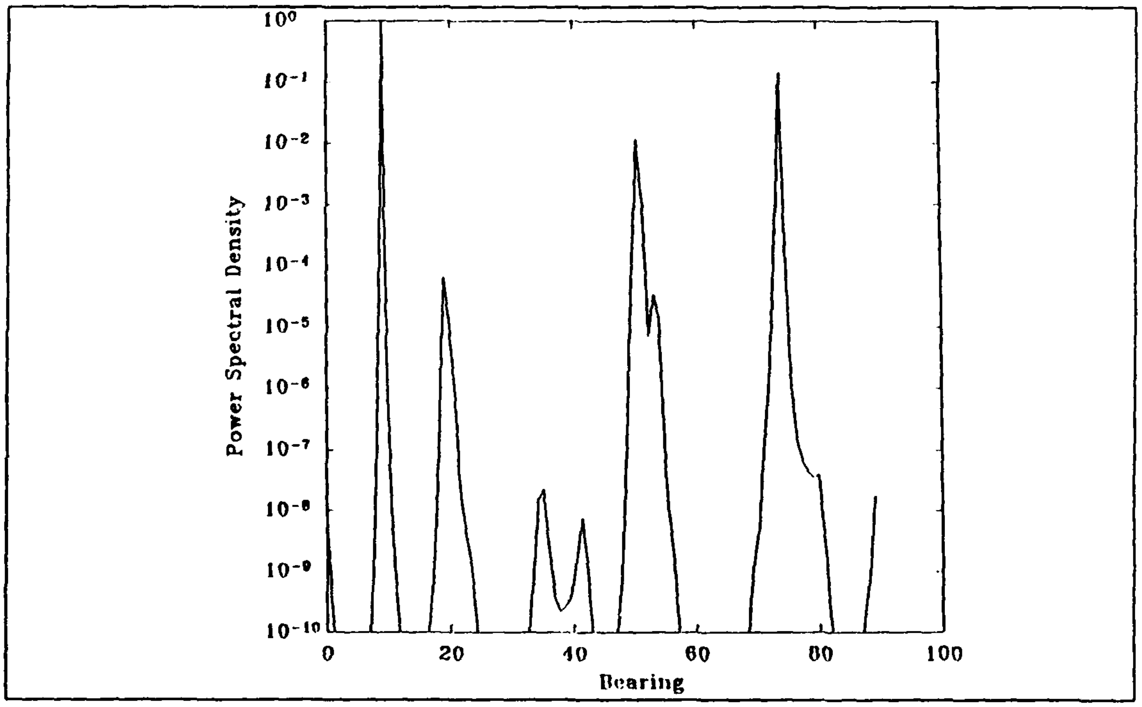

Figure 5. One target at 9° (–5dB) : product of the PSDs 5 eigenvectors

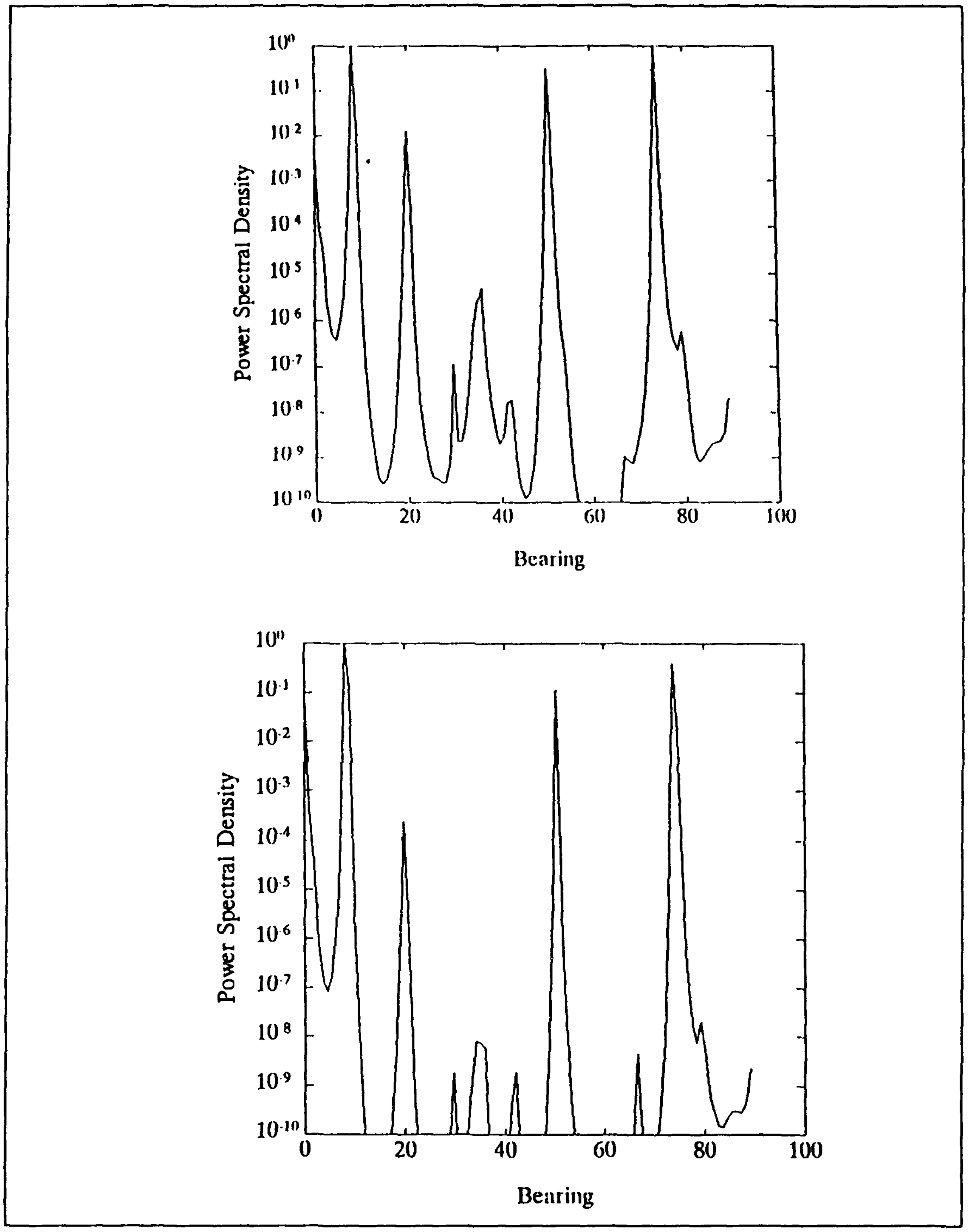

Figure 6. One target at 9° (–10 dB) : product of the PSDs

(a) 5 eigenvectors (b) 7 eigenvectors

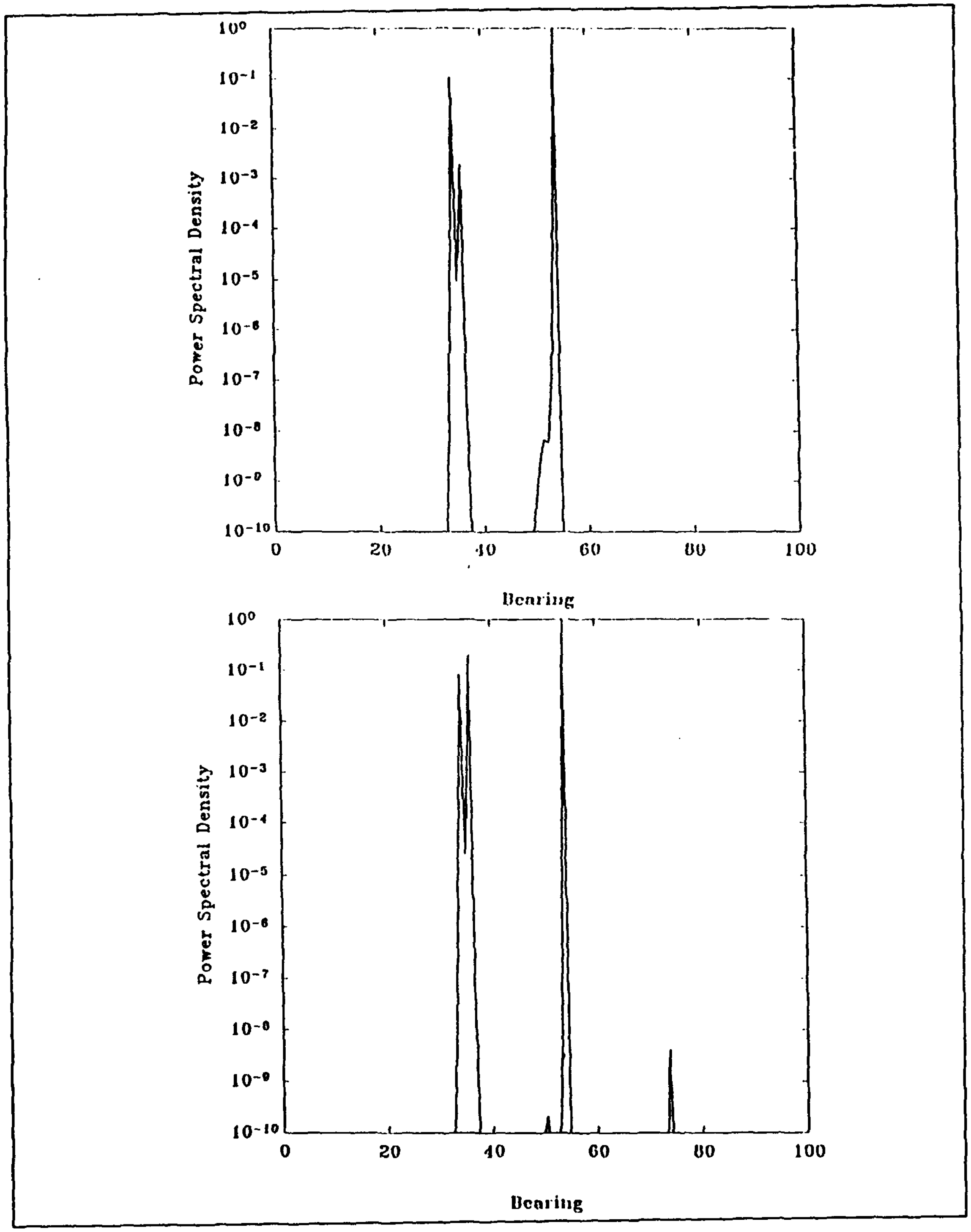

Figure 7. **Three targets at 34°, 36° and 54° (0 dB) :** product of the PSDs
5 eigenvectors (a) Ritz vector (b) Rayleigh quotient iteration

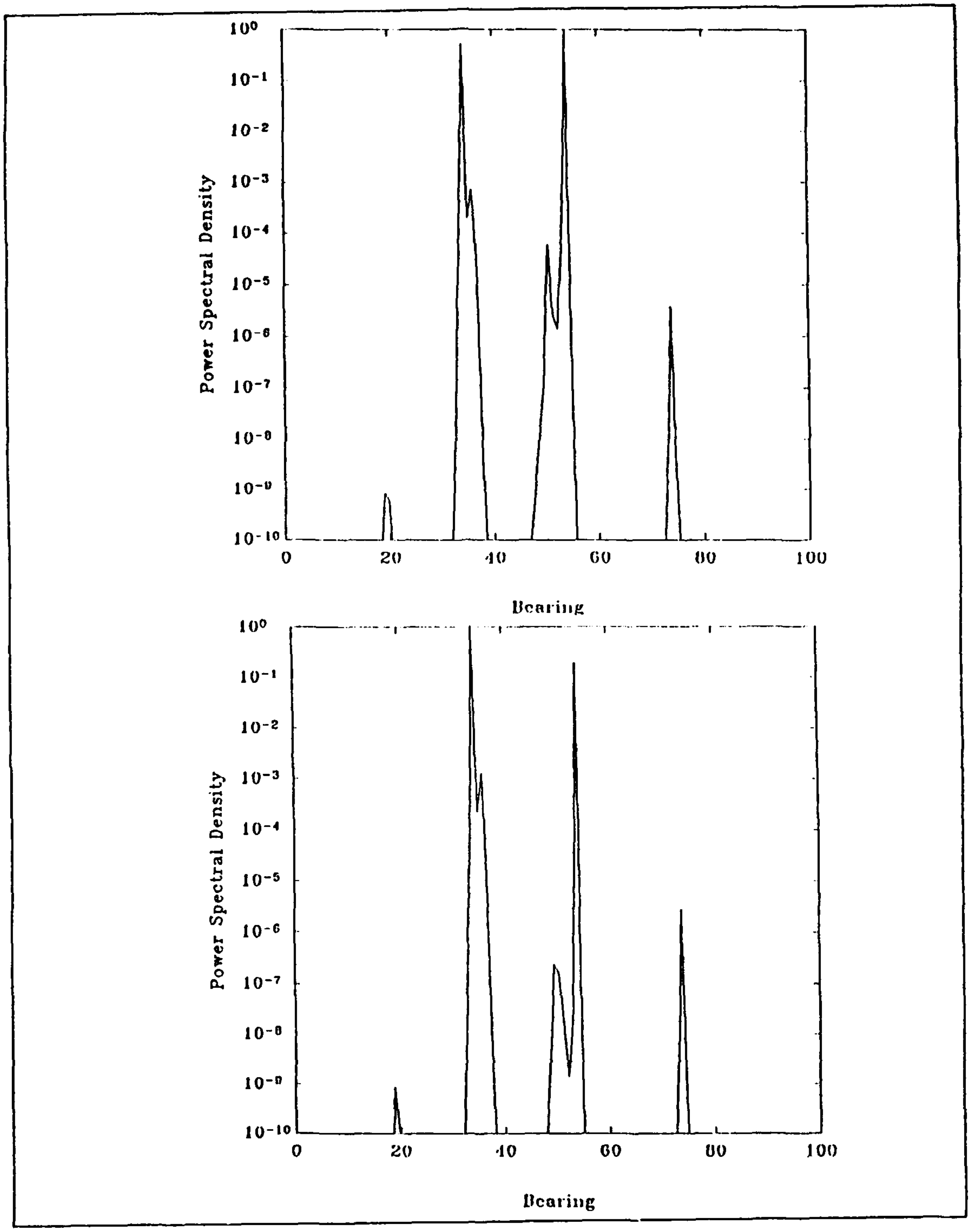

Figure 8. Three targets at 34°, 36° and 54° (–5 dB) :

(a) no reorthogonalizing (b) with complete reorthogonalizing

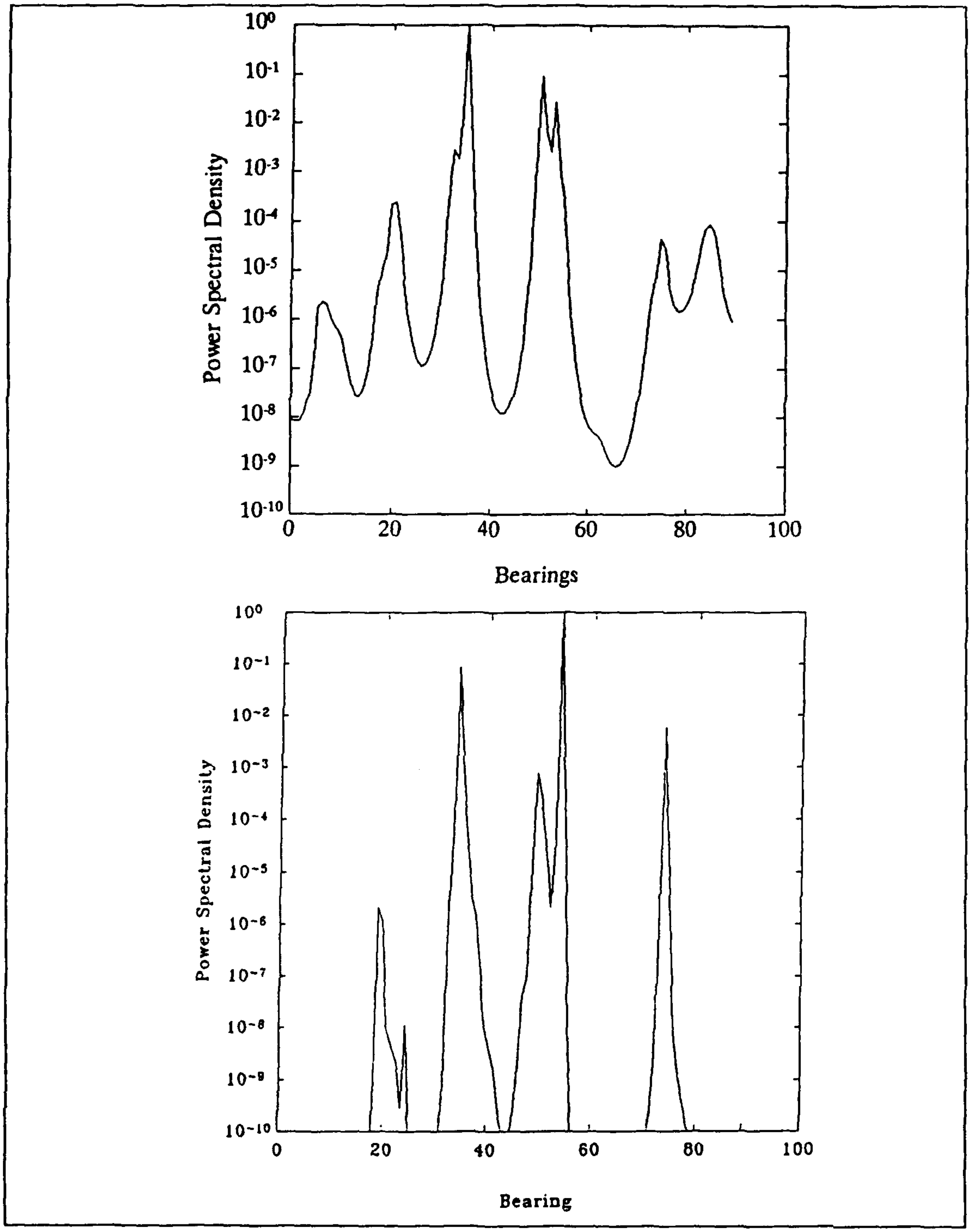

Figure 9. Three targets at 34°, 36° and 54° (−10 dB) :
product of the PSDs (a) 3 eigenvectors (b) 5 eigenvectors

III. BLOCK LANCZOS ALGORITHM

In the previous chapter we presented the single vector Lanczos method. The single vector method can be used to find a few extreme eigenvalues of any given real symmetric matrix R. However, this method does not determine the multiplicities of the eigenvalues directly. Besides, it does not determine a complete basis for the invariant subspace corresponding to any such multiple eigenvalue. Here we consider an alternative approach which directly determines the multiplicities of the eigenvalues and the corresponding eigenvectors [Ref. 1].

The method known as the block Lanczos algorithm is an extension of the Lanczos algorithm in which a *block* of vectors rather than a single vector is iterated [Ref. 4]. We produce a block tridiagonal matrix in place of the usual tridiagonal matrix produced in the single vector Lanczos method. The block Lanczos method can be used in a manner proposed by Paige [Ref. 10]. That is, one can compute a sequence of estimates of the eigenvalues of the matrix R from the block tridiagonal matrix.

Several researchers have worked on the block Lanczos algorithm, in particurlar Kahan and Parlett [Ref. 6], Cullum and Willoughby [Ref 1], and Golub and Underwood [Ref. 4]. In this chapter we describe the basic idea of the block Lanczos method, develop the algorithm, present several simulation results, and compare them with those of the single vector algorithm.

A. BLOCK LANCZOS METHOD

Given a large real correlation matrix R of size n×n, we can generate a banded tridiagonal matrix T_s of size qs×qs, where qs<n, q is the block size, and s is the number of the Lanczos blocks. Starting from an initial n×q orthonormal matrix $\mathbf{Q}_1$, the purpose is to compute a sequence of mutually orthonormal n×q matrices $\mathbf{Q}_2$, $\mathbf{Q}_3$, $\mathbf{Q}_4, \cdots, \mathbf{Q}_s$ such that the space of vectors spanned by the columns of these matrices contains the columns of the matrices $\mathbf{Q}_1$, $R\mathbf{Q}_1$, $R^2\mathbf{Q}_1$, ..., $R^{s-1}\mathbf{Q}_1$, where $0<q\leq\frac{n}{2}$ and $1<s\leq\frac{n}{q}$. Note that usually we have ps<<n. The block Lanczos algorithm can be summarized as follows [Ref. 1]: For i=1,2,···,s, compute

$$A_i = \mathbf{Q}_i^T(R\mathbf{Q}_i - \mathbf{Q}_{i-1}B_i^T) \tag{3–1}$$

$$P_i = R\mathbf{Q}_i - \mathbf{Q}_iA_i - \mathbf{Q}_{i-1}B_i^T \tag{3–2}$$

$$\mathbf{Q}_{i+1}B_{i+1} = P_i \quad (\text{QR } \textit{factorization of } P_i). \tag{3–3}$$

In this procedure, $\mathbf{Q}_{i+1}$ is orthonormal to all previous $\mathbf{Q}_i$. The purpose of the block Lanczos algorithm is to find A_i, B_{i+1} and $\mathbf{Q}_{i+1}$, where A_i and B_{i+1} are the element matrices of the desired tridiagonal matrix, and $\mathbf{Q}_{i+1}$ is the $(i+1)^{st}$ orthonormal Lanczos block. The blocks $\mathbf{Q}_i$, for i=1,2,···,s, form an orthonormal basis of the Krylov subspace, defined as

$$\mathcal{K}^s(\mathbf{Q}_1,R) \equiv \text{span}\{\mathbf{Q}_1,R\mathbf{Q}_1,R^2\mathbf{Q}_1, \cdots, R^{s-1}\mathbf{Q}_1\} \tag{3–4}$$

corresponding to the first block $\mathbf{Q}_1$. The matrix $\mathbf{Q}$, defined as a catenation of the

Lanczos blocks

$$\mathbf{Q} = [\mathbf{Q}_1\, \mathbf{Q}_2\, \mathbf{Q}_3 \cdots \mathbf{Q}_s] \tag{3–5}$$

is orthogonal [Ref. 4], *i.e.*, $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ or $\mathbf{Q}^T = \mathbf{Q}^{-1}$. We then generate the banded tridiagonal matrix T_s using A_i and B_i:

$$T_s = \begin{bmatrix} A_1 & B_2^T & & & & \\ B_2 & A_2 & B_3^T & & \mathbf{0} & \\ & B_3 & A_3 & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & \mathbf{0} & & & \cdot & B_s^T \\ & & & & B_s & A_s \end{bmatrix}. \tag{3–6}$$

The elements A_i and B_i are q×q coefficient matrices, so that T_s is a qs×qs banded matrix. The off-diagonal blocks, B_i, are upper triangular matrices and the main diagonal blocks, A_i, are symmetric matrices so that T_s itself is a symmetric matrix. Also, the banded tridiagonal matrix T_s can be determined by

$$T_s = \mathbf{Q}^T R \mathbf{Q}. \tag{3–7}$$

There are basically two different types of block Lanczos procedures, namely, the iterative procedure and the noniterative procedure. The noniterative procedure follows along the lines of the single vector Lanczos procedure. In this method a sequence of blocks $\{\mathbf{Q}_1, \mathbf{Q}_2, \cdots\}$ are generated where the length of the sequence can be determined by the size of T_s and the amount of storage available. The generated

Lanczos blocks may or may not be reorthogonalized. The iterative procedure uses the block recursion to generate the block tridiagonal matrix. First, the relevant eigenvalues and eigenvectors of these block tridiagonal matrices are computed. Then the corresponding Ritz vectors are computed and used as updated approximations to the desired eigenvectors. If convergence has not been achieved in k iterations (see Step 2 below), more iterations are performed these updated eigenvectors until the procedure has converged.

The following steps show the basic iterative block Lanczos procedure:

Step 1. For k=1 start with an initial arbitrary $n \times q$ block $\mathbf{Q}_1^k$ where the columns of Q_1^k are orthonormal.

Step 2. Compute $\mathbf{P}_1^k = R\mathbf{Q}_1^k - \mathbf{Q}_1^k A_1^k$ using $\mathbf{Q}_1^k$ where $A_1^k = (\mathbf{Q}_1^k)^T R\mathbf{Q}_1^k$ and use the norms of the columns of $\mathbf{P}_1^k$ to check for convergence. If convergence has occured, then stop; otherwise, go to Step 3.

Step 3. Generate a sequence of blocks $\mathbf{Q}_j^k$ using the recursion in Eqn(3–2) and Eqn(3–3) for $j=2, 3, \cdots, s$. Use the coefficient matrices A_j^k and B_{j+1}^k to define the real symmetric block tridiagonal matrix T_s^k.

Step 4. Compute the q algebraically smallest eigenvalues of T_s^k and the correspoding eigenvectors $\mathbf{Y}^k$ where $\mathbf{Y}^k = \{\mathbf{y}_1^k, \mathbf{y}_2^k, \cdots, \mathbf{y}_q^k\}$.

Step 5. Obtain the new Lanczos block $\mathbf{Q}_1^{k+1}$ given by

$$\mathbf{Q}_1^{k+1} = \mathbf{Q}^k \mathbf{Y}^k \tag{3–8}$$

where $\mathbf{Q}^k = \{\mathbf{Q}_1^k, \mathbf{Q}_2^k, \cdots, \mathbf{Q}_s^k\}$. Increment k to k+1 and go to Step 2.

From the above procedure we can generate the block tridiagonal matrix T_s and also compute the eigenvectors Y^k which approximate the eigenvectors of R. As noted, Step 2 provides a check for the convergence of the procedure.

If it were the case that qs = n, then T_s would be similar to R and the eigenvalues of T_s would also be similar to the eigenvalues of R. Particularly, some of the extreme eigenvalues of T_s would be approximately the same as the corresponding eigenvalues of R. Generally, because of the numerical properties of the block Lanczos algorithm, it is not practical to carry the method through to completion [Ref. 1]. The importance of the algorithm lies in the fact that some of the smallest (and largest) eigenvalues of T_s will closely approximate the corresponding eigenvalues of R for values of s such that qs << n. This is stated by the following theorem.

Theorem 3.1 [Ref. 1]. Let $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ be the eigenvalues of R and $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n$ be the corresponding orthonormal eigenvectors. Assume that $\lambda_q < \lambda_{q+1}$. Apply the block Lanczos recursion in Eqns(3–1)–(3–3) to R generating s blocks and let $\mu_1 \le \mu_2 \le \cdots \le \mu_n$ be the eigenvalues of T_s. Suppose that

$$W \equiv \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} \equiv \mathbf{V}^T \mathbf{Q}_1 \tag{3–9}$$

is the n×q matrix of projections of the starting block of vectors $\mathbf{Q}_1$ on the eigenvectors of R, where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \cdots, \mathbf{v}_n]$ and W_1 is a q×q matrix composed of the first q rows of W. Suppose further that W_1 is nonsingular so that σ_{min}, the smallest singular value of W_1, is greater than zero. Then for k = 1, 2, 3, ..., q, the eigenvalues of T_s satisfy

$$\lambda_k \le \mu_k \le \lambda_k + \epsilon_k^2 \tag{3–10}$$

where the spread ϵ_k^2 is given by

$$\epsilon_k^2 = \frac{(\lambda_n - \lambda_k)\tan^2\theta}{\mathfrak{F}_{s-1}^2\left(\dfrac{1+\gamma_k}{1-\gamma_k}\right)} \tag{3–11}$$

and $\theta = \cos^{-1}\sigma_{min}$, $\gamma_k = (\lambda_k - \lambda_{q+1})/(\lambda_k - \lambda_n)$, and $\mathfrak{F}_{s-1}$ is the $(s-1)^{th}$ Chebyshev polynomial of the first kind.

This theorem illustrates the importance of the local gaps $|\lambda_k - \lambda_{q+1}|$, but does not show the potential positive effect of the outer loop iteration of an iterative block Lanczos procedure on reducing the overall effective spread and thereby improving the convergence rate.

Note that, as we have defined it, the block Lanczos method is not a method for finding the eigenvalues and the eigenvectors of a symmetric matrix R. Rather, it is a procedure for finding a block tridiagonal matrix T_s which is similar to R. To produce a complete algorithm for finding the eigenvalues and the eigenvectors, we need to combine the Lanczos algorithm with a technique for finding the eigenvalues μ_k and the eigenvectors $\mathbf{y}_k$ of T_s such as the QR algorithm.

Now, we will consider certain properties of the block Lanczos algorithm and problems associated with its implementation and application.

The computed Lanczos blocks $\mathbf{Q}_i$ are desired to be mutually orthogonal. In practice, however, because of the arithmetic errors when $\mathbf{P}_i$ is computed, they rapidly lose orthogonality. Thus, after a few iterations of the block Lanczos algorithm, the current $\mathbf{Q}_i$ is no longer orthogonal to the previous Lanczos blocks $\mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_{i-1}$. The subsequent losses in orthogonality between the blocks caused by the roundoff errors increase as we increase the number of Lanczos blocks. At some stage when these losses have accumulated sufficiently, the assumption that the block tridiagonal Lanczos matrix is the projection of the given matrix R on the subspaces **Q** will be false [Ref. 1]. As a result the q smallest eigenvalues of T_s may not approximate the q smallest eigenvalues of R. Thus, it requires costly reorthogonalization of each $\mathbf{Q}_{i+1}$ with respect to all the previous Lanczos blocks to maintain the stability of the algorithm [Ref. 4].

Loss of orthogonality goes hand-in-hand with the convergence of some of the eigenvalues of T_s to the eigenvalues of R. In this case we have two options, stop the Lanczos iterations as the blocks begin to lose their mutual orthogonality or reorthogonalize the blocks if more iterations are desired. The difficulty in using the Lanczos method in this way lies in reliably and efficiently determining at what point the orthogonality is being lost. In order to reorthogonalize all Lanczos blocks, we first reorthogonalize the residual matrix P_i with respect to all the previous Lanczos blocks and compute the next block $\mathbf{Q}_{i+1}$ and the coefficient matrix B_{i+1} such that $P_i=\mathbf{Q}_{i+1}B_{i+1}$. This modification preserves the stability of the algorithm but at a considerable cost because the reorthogonalization process requires a large number of arithmetic operations. The need for reorthogonalization seems to increase with the size n of R and the number of blocks that are required to be computed in the algorithm [Ref. 1].

We now describe how to choose the block size q. It is usually best to choose q equal to the number of eigenvalues and eigenvectors r that we are attempting to compute which could be the multiplicity of the smallest eigenvalue. Theorem 3.1 suggests that a good choice for q is one for which the gap between λ_q and λ_{q+1} is fairly large.

B. ALGORITHM

It is possible to find a few of the extreme eigenvalues and corresponding eigenvectors of a real symmetric matrix using the block Lanczos algorithm rather than computing the entire matrix decomposition. Each of these smallest eigenvalues and the corresponding eigenvector of the autocorrelation matrix for received signals from a sensor array has the spectral information to estimate the direction-of-arrival.

The computer simulation experimental set up used is the same as the one shown in Figure 1. We receive the signals at a linear array containing M equally-spaced sensors and generate the n×n autocorrelation matrix of these received signals.

1. Reduction

The reduction of the data proceeds as follows. Using the block Lanczos algorithm we reduce the autocorrelation matrix R into a block tridiagonal matrix that has the same extreme eigenvalues as R. *In this section* we will present a practical algorithm to implement the block Lanczos method.

Given an n×n autocorrelation matrix R, we generate an initial n×q matrix $\mathbf{Q}_1$ which is chosen arbitrarily and orthonormalized. The number of vectors in each block, q, is considered to be between 3 and 5 in this study. To begin, we compute $\mathbf{RQ}_1$ and a residual matrix $\mathbf{P}_1$ given by

$$\mathbf{P}_1 = \mathbf{RQ}_1 - \mathbf{Q}_1 \mathrm{A}_1 \tag{3–15}$$

where A_1 is a q×q coefficient matrix chosen so that the Euclidean norm of $\mathbf{P}_1$ is minimized with respect to all possible choices of A_1 [Ref. 4]. It can be shown [Ref. 1] that $\|\mathbf{P}_1\|$ is minimized when

$$A_1 = \mathbf{Q}_1^T R \mathbf{Q}_1. \tag{3–16}$$

The q×q matrix A_1 forms the first block on the main-diagonal of the block tridiagonal matrix, T_s (see Eqn(3–6)). With this choice for A_1, we have

$$\mathbf{P}_1 = (\mathbf{I} - \mathbf{Q}_1\mathbf{Q}_1^T) R \mathbf{Q}_1. \tag{3–17}$$

That is, $\mathbf{P}_1$ is the projection of $R\mathbf{Q}_1$ onto the space orthogonal to that spanned by the columns of $\mathbf{Q}_1$. The second Lanczos block of vectors $\mathbf{Q}_2$ and a q×q upper triangular coefficient matrix B_2 are then obtained by using the QR factorization with modified Gram-Schmidt procedure on $\mathbf{P}_1$:

$$\mathbf{Q}_2 B_2 = \mathbf{P}_1. \tag{3–18}$$

The current Lanczos block $\mathbf{Q}_2$ is orthonormal to the previous block $\mathbf{Q}_1$. The upper triangular matrix B_2 and its transposed version B_2^T form the first elements in the sub-diagonal and the super-diagonal, respectively, of the block tridiagonal matrix T_s.

The remaining matrices in the sequence of the Lanczos blocks $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\cdots$, $\mathbf{Q}_s$, where $s<<n$, are computed as follows: For $i=2,3,\cdots,s$, compute

$$A_i = \mathbf{Q}_i^T (R\mathbf{Q}_i - \mathbf{Q}_{i-1} B_i^T) \tag{3–19}$$

$$\mathbf{P}_i = R\mathbf{Q}_i - \mathbf{Q}_i A_i - \mathbf{Q}_{i-1} B_i^T \tag{3–20}$$

$$\mathbf{Q}_{i+1} B_{i+1} = \mathbf{P}_i \tag{3–21}$$

where $\mathbf{Q}_{i+1}$ and B_{i+1} are obtained as the QR factorization of the residual matrix $\mathbf{P}_i$. A modified Gram–Schmidt procedure can be used to reorthgonalize the columns of $\mathbf{P}_i$. This means that $\mathbf{Q}_{i+1}$ is orthonormal to all previous matrices $\mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_i$. As we increase the number of the Lanczos blocks, their mutual orthogonality is preserved because of the built-in QR algorithm to factor the residual matrix $\mathbf{P}_i$. Consequently, the space spanned by $\mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_s$ contains the columns of the matrices $\mathbf{Q}_1, R\mathbf{Q}_1, R^2\mathbf{Q}_1, \cdots, R^{s-1}\mathbf{Q}_1$ which form the orthonormal basis of the Krylov subspace. Thus, we do not need to reorthogonalize the Lanczos blocks when we use the algorithm in Eqns(3–17)–(3–19).

2. Eigendecomposition

We need to compute the eigenvalues of the block tridiagonal matrix T_s which approximate the eigenvalues of an autocorrelation matrix. Then the eigenvectors of R corresponding to these eigenvalues are determined by knowing the matrix of Lanczos blocks.

There are several techniques for computing the eigenvalues and eigenvectors of a given matrix. The fundamental algebraic eigenproblem is to determine the eigenvalues μ_i given the set of qs homogeneous linear equations in qs unknowns [Ref. 17]

$$T_s \mathbf{y}_i = \mu_i \mathbf{y}_i \qquad \text{for } i=1,2,\cdots,qs \tag{3–22}$$

where the eigenvectors $\mathbf{y}_i$ of T_s satisfy

$$\mathbf{y}_i^T\mathbf{y}_j = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j, \end{cases} \tag{3–23}$$

that is, $\mathbf{y}_i$ are mutually orthonormal vectors. From Eqn(3–22) the characteristic equation associated with the matrix T_s is given by

$$\det(T_s - \mu_i I) = 0. \tag{3–24}$$

Expanding the determinant, we have the polynomial equation

$$\alpha_0 + \alpha_1\mu + \cdot\cdot\cdot + \alpha_{qs\text{-}1}\mu^{qs\text{-}1} + \mu^{qs} = 0 \tag{3–25}$$

where α_j are the coefficients of μ^j and the roots of this polynomial give us the eigenvalues μ_i of the T_s. Corresponding to any eigenvalue μ_i, Eqn(3–22) has at least one non-trivial solution $\mathbf{y}_i$. Since the eigenvalues of T_s approximate a few of the extreme eigenvalues of autocorrelation matrix R, we choose the q smallest eigenvalues and the corresponding eigenvectors of T_s and use the Ritz vector to compute the eigenvectors of R given by [Ref. 1]

$$\mathbf{X} = \mathbf{QY} \tag{3–26}$$

where $\mathbf{X}$ is a group of approximate eigenvectors of R, $\mathbf{X}=[\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_q]$, $\mathbf{Q}$ is a matrix of Lanczos blocks, and $\mathbf{Y}$ is a group of q smallest eigenvectors of T_s, $\mathbf{Y}=[\mathbf{y}_1\ \mathbf{y}_2 \cdots \mathbf{y}_q]$. Also, we can use the Rayleigh quotient iteration to find the eigenvectors of R related to the eigenvalues of T_s. However, this method is slower than using the Ritz vector because it needs more time to converge for a given eigenvalue.

Now, using the eigenvectors of the autocorrelation matrix R we estimate the spatial spectrum of the received signals. Each of the eigenvectors contains the true spectral information as well as some spurious peaks. The direction-of-arrival of point sources can be estimated by computing the spatial power spectral density of the eigenvectors of R. The power spectral density estimate for the j^{th} eigenvector corresponding to one of the smallest eigenvalues of R is computed as follows:

$$S^{j}_{xx}(f) = \left| \frac{1}{\sum_{i=0}^{n-1} x_{ji} z^{-i}} \right|^{2}_{z=e^{j2\pi f}} \tag{3–27}$$

where x_{ji} are the elements of the j^{th} eigenvector, $\mathbf{x}_j$ and $0 \leq f \leq 0.5$.

C. SIMULATION RESULTS

Using the block Lanczos algorithm we can selectively compute a few of the smallest eigenvalues and eigenvectors of an autocorrelation matrix. These eigenpairs are in the noise subspace and contain the spectral information of the source bearings from an array of sensors. Thus, we could estimate the spatial power spectral density for each eigenvector using Eqn(3–27). We have used the spectral product of several individual PSDs to improve the direction-of-arrival estimation performance. The advantage in using the multiplicative PSD function was demonstrated in Chapter 2 for the single vector Lanczos algorithm. The same advantage holds for the block Lanczos algorithm.

The computer simulation experiment consists of an equally-spaced linear array of 100 sensors receiving signals of known temporal frequency from various bearings. The size of the autocorrelation matrix computed is 25×25. The number of smallest eigenpairs to be estimated is q, which is the size of the Lanczos block. We choose q to be 3, 5, or 7 in these examples and used 5 dB, 0 dB, –5 dB and –10 dB signal-to-noise ratios (SNR).

Example 1 is the detection of a target at 9^o for different SNRs. Figure 10 shows the spectral overlay of 5 eigenvectors corresponding to the smallest eigenvalues at an SNR of 5 dB. The spectrum of each eigenvector has a common peak at the true bearing 9^o. Figure 11(a) and Figure 11(b) show improved DOA estimation where the former figure illustrates the average of 5 eigenvectors and the latter figure shows the spectral multiplication of those eigenvectors. As can be seen, the spectral multiplication technique has greatly improved the nulls and the peak at 9^o. Thus, in the remainder of the results, we have used the spectral multiplication method to compute the PSD function. Figure 12(a) shows the performance at 0 dB using 3 eigenvectors and Figure 12(b) shows the result at an SNR of 0 dB using 5 eigenvectors. Improved performance is obtained by using more eigenvectors. In Figure 12(a) we have several large spurious peaks around the true peak at 9^o, but in Figure 12(b) we have just 2 small spurious peaks and the true bearing is clearly evident. Figure 13(a) and Figure 13(b) show the results at an SNR of –5 dB using 3 and 5 eigenvectors, respectively. The spurious peaks have increased magnitudes but the difference between the true peak and the spurious peaks is still large enough to determine the true bearing. Figure 14(a) and Figure 14(b) show the result at an SNR of –10 dB. Using 3 eigenvectors (Figure 14(a)), it is hard to determine the true bearing. Using 5 eigenvectors (Figure 14(b)), the true peak appears to be larger than the spurious peaks and we can recognize the true bearing.

In Example 2 we have 3 targets at 34°, 36° and 54°. Two targets are very closely spaced in bearing. At 0 dB, the results of two cases, where one has used 5 eigenvectors (Figure 15(a)) and the other 7 eigenvectors (Figure 15(b)), indicate good performance. There are no spurious peaks and the resolution is clearly acheived. Figure 16(a) shows the result at –5 dB using 5 eigenvectors. In this result, the resolution is acheived but one spurious peak is as large as the true peaks. In Figure 16(b), the result shows improved performance using 7 eigenvectors. There are no spurious peaks and good spectral resolution is acheived. At –10 dB, the algorithm cannot separate the two targets located at 34° and 36° when we use 5 eigenvectors (Figure 17(a)). When we use 7 eigenvectors however (Figure 17(b)) it can separate the two closely located sources. Nevertheless, a number of spurious peaks are larger in magnitude than the peak at 36° making it impossible to determine the true bearings accurately.

The results in this chapter show that the eigenvectors computed using the block Lanczos algorithm can be used to determine the spectrum even in very low SNRs. Since the block Lanczos method can compute a few of the extreme eigenpairs of a large symmetric matrix, it is efficient in computations and storage.

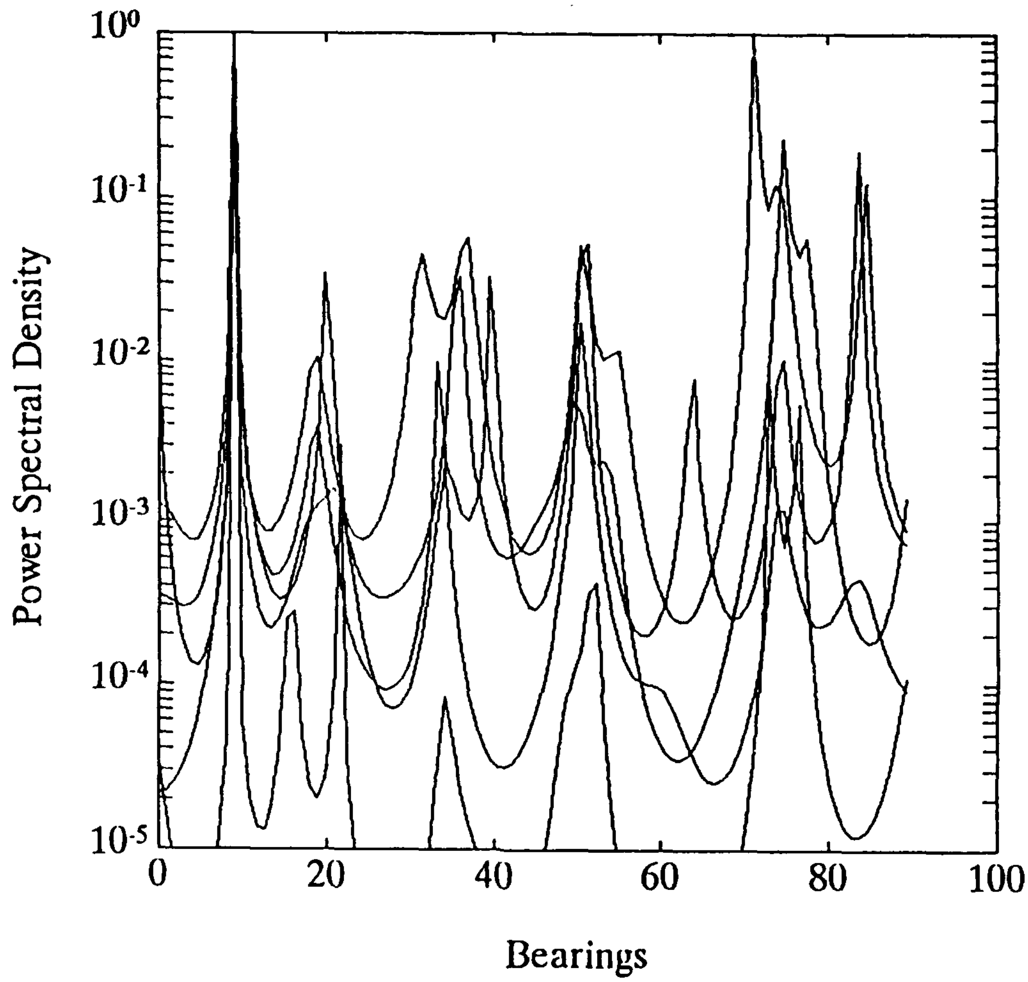

Figure 10. **One target at 9° (5 dB)** : overlay of 5 eigenvectors

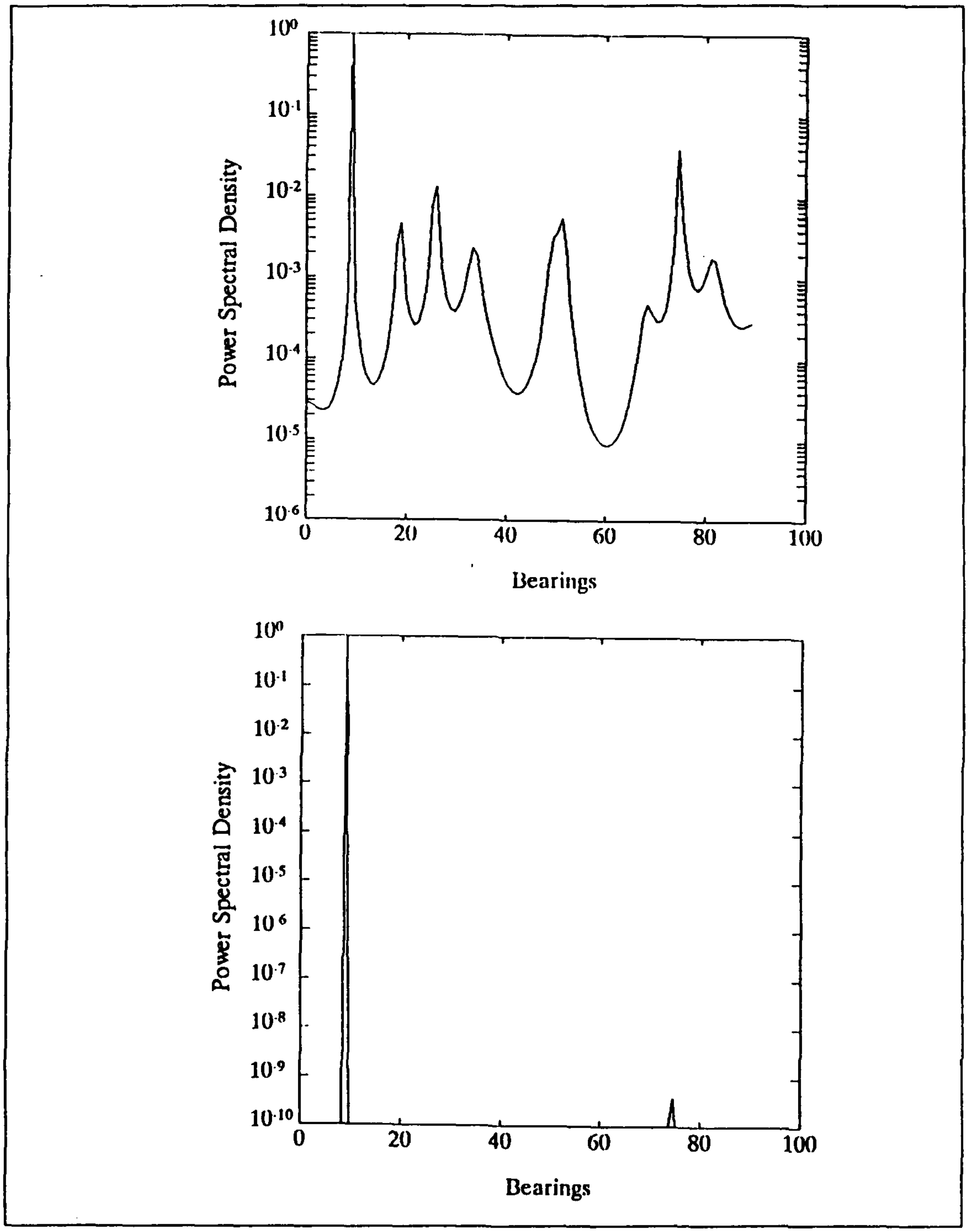

Figure 11. One target at 9° (5 dB) : (a) average of 5 eigenvectors
(b) product of the PSDs 5 eigenvectors

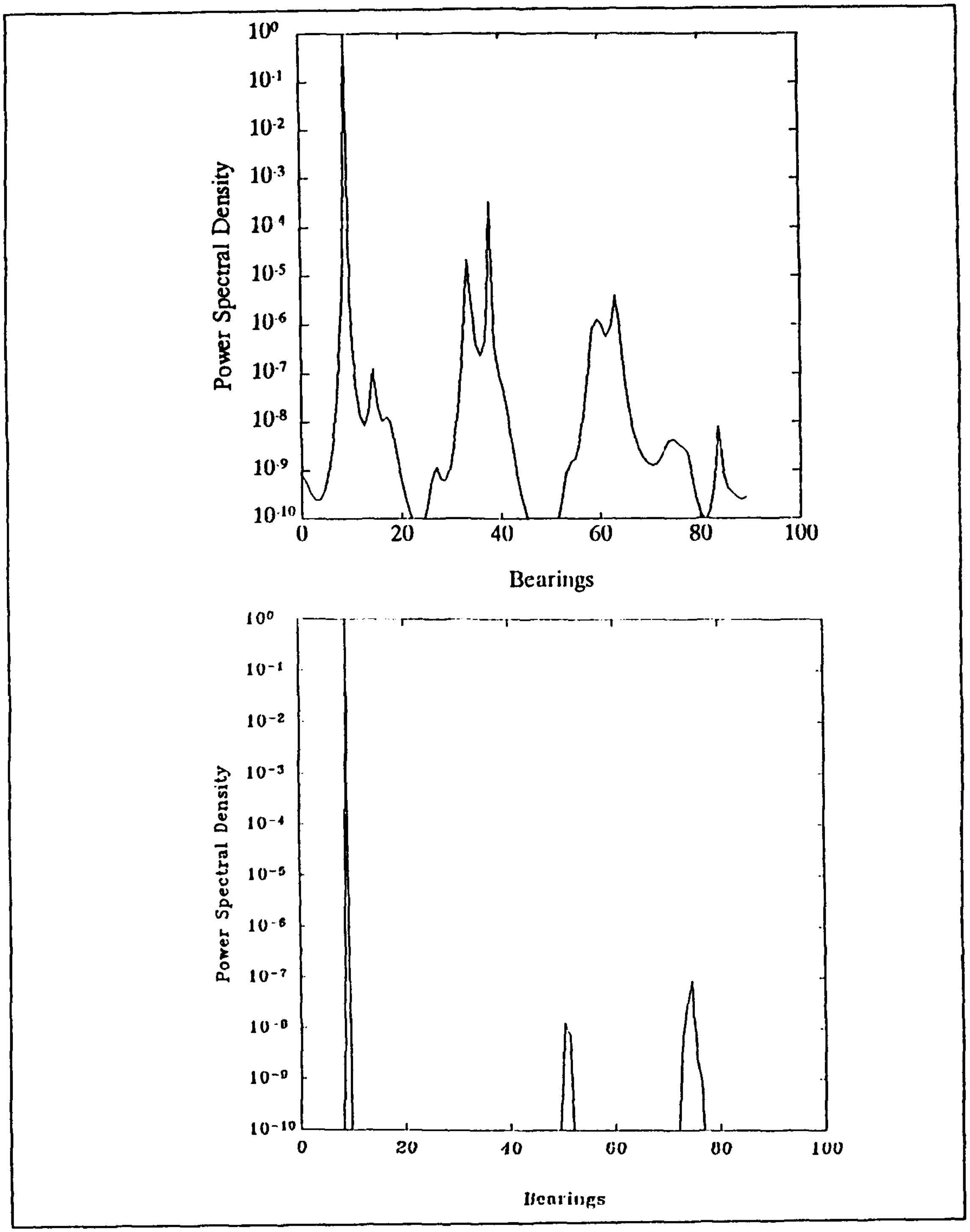

Figure 12. **One target at 9° (0 dB)** : product of the PSDs

(a) 3 eigenvectors (b) 5 eigenvectors

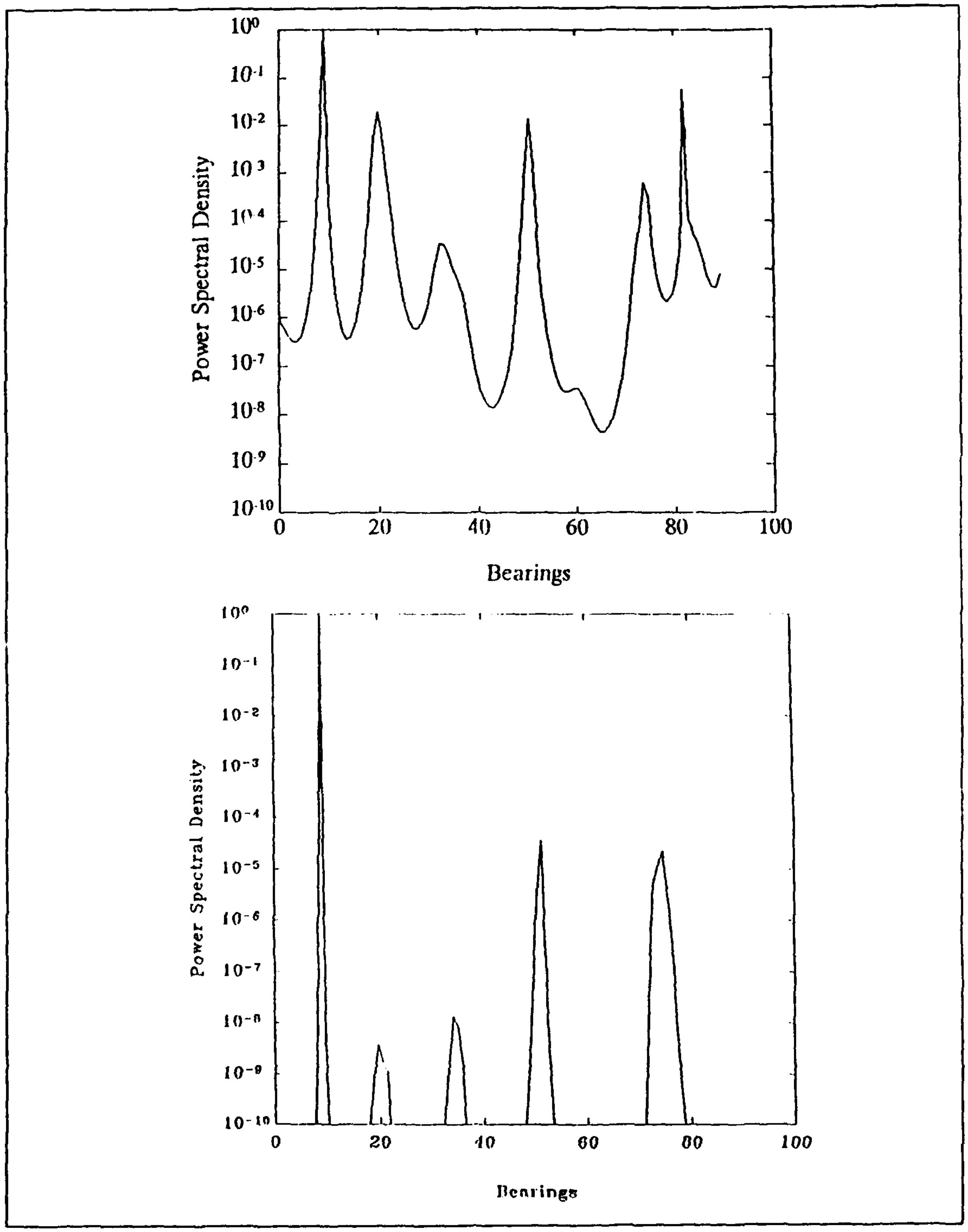

Figure 13. **One target at 9° (–5 dB) : product of the PSDs**

(a) 3 eigenvectors (b) 5 eigenvectors

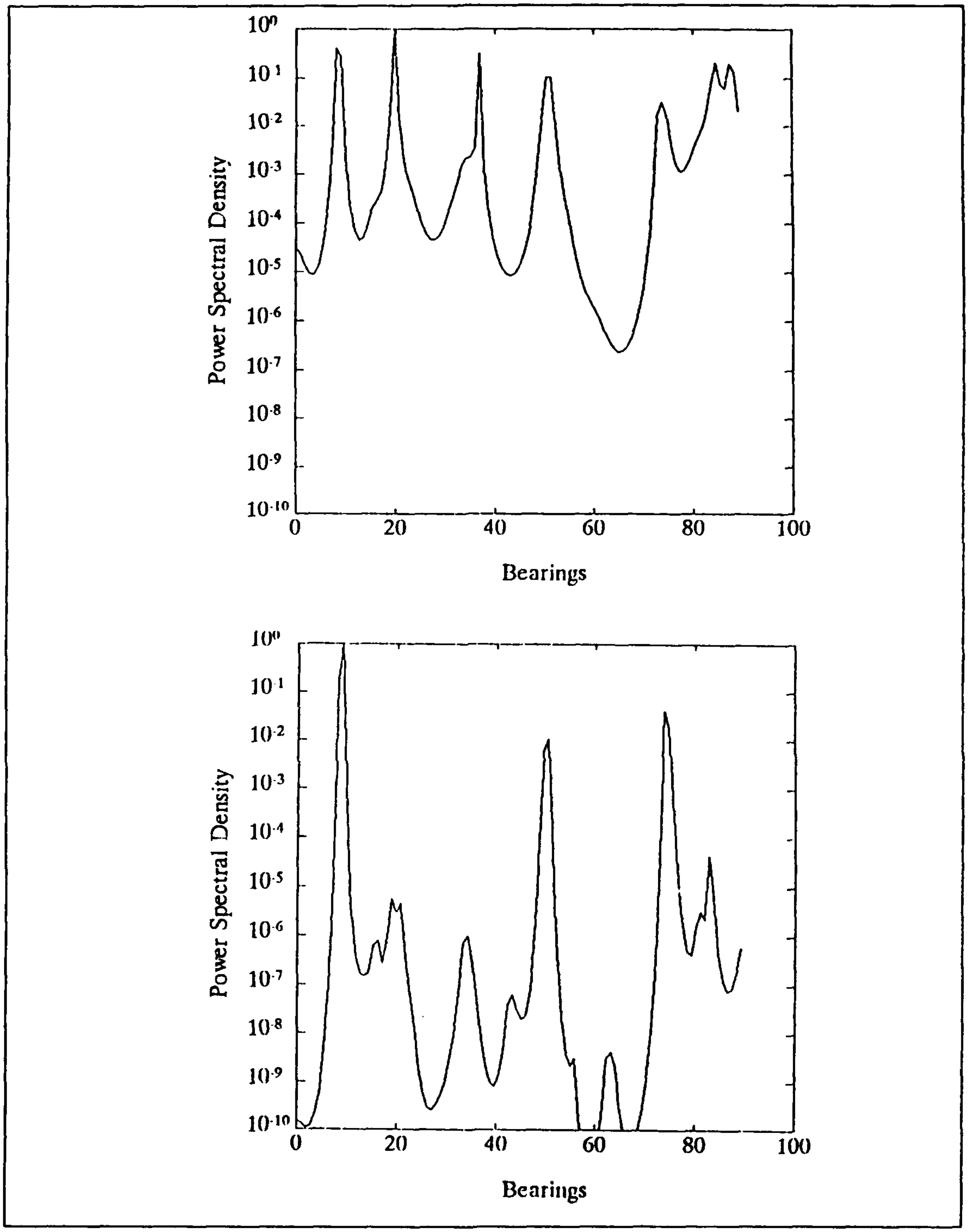

Figure 14. **One target at 9° (–10 dB)** : product of the PSDs

(a) 3 eigenvectors (b) 5 eigenvectors

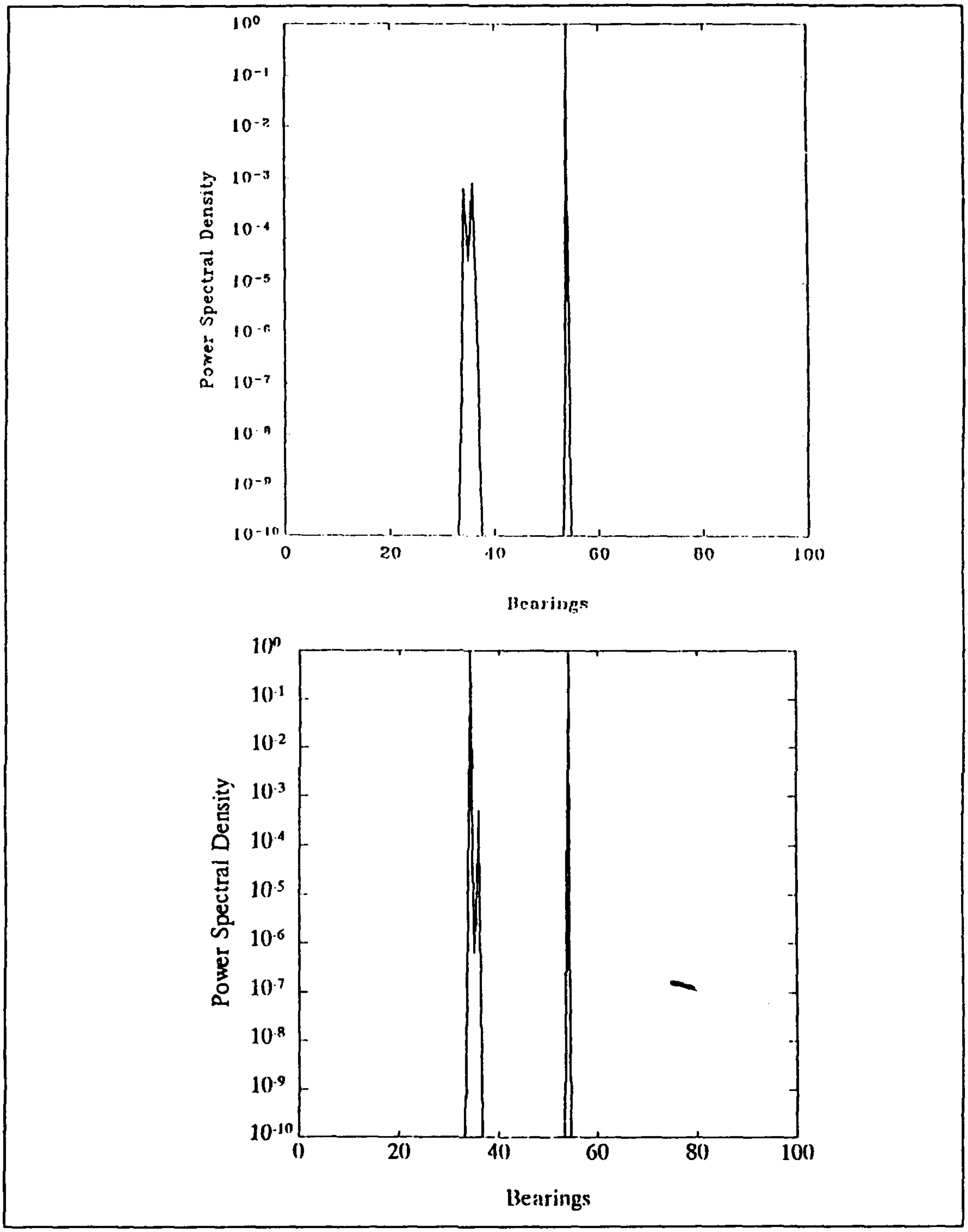

Figure 15. **Three targets at 34°, 36° and 54° (0dB) :**

product of the PSDs (a) 5 eigenvectors (b) 7 eigenvectors

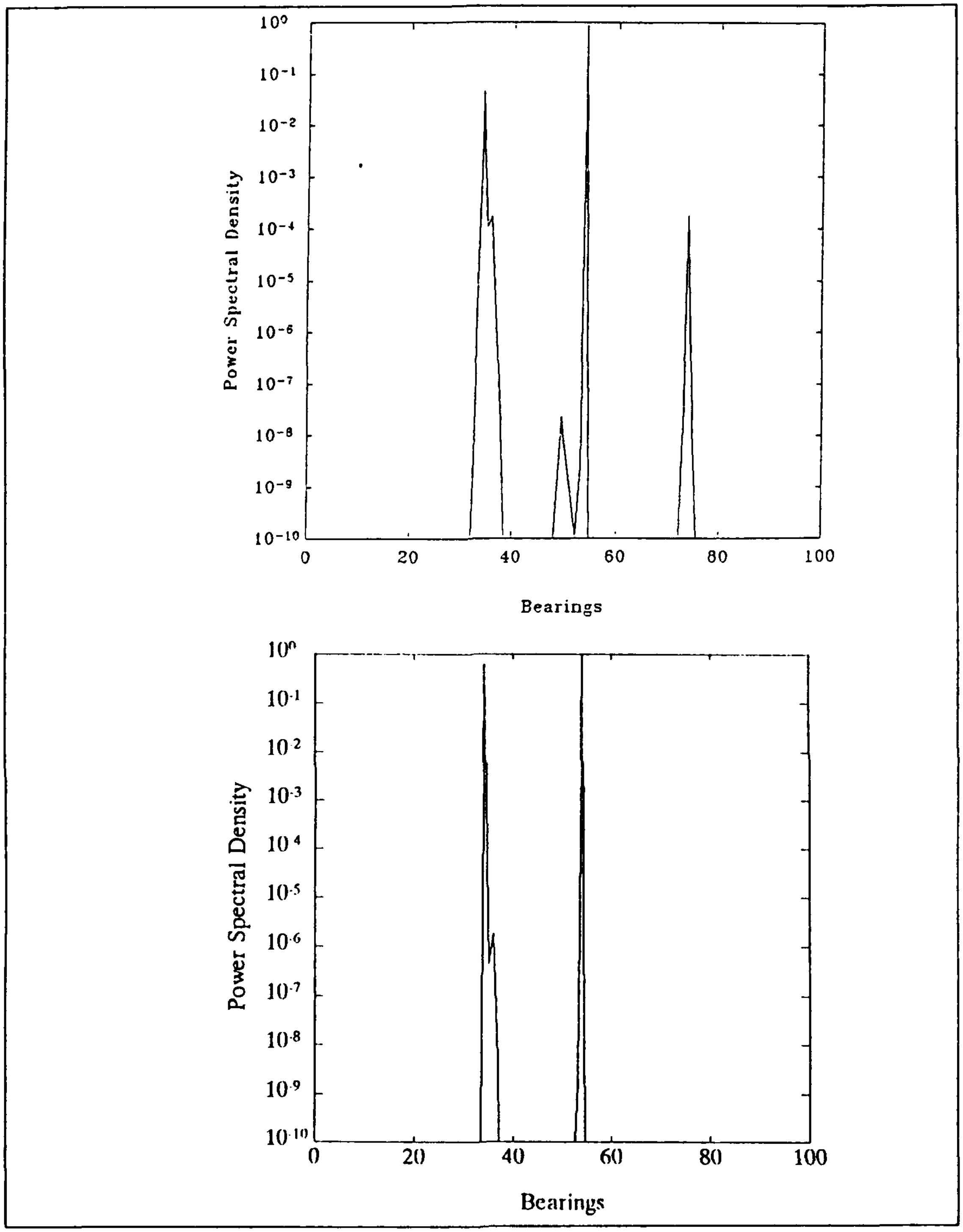

Figure 16. **Three targets at 34°, 36° and 54° (–5 dB) :**

product of the PSDs (a) 5 eigenvectors (b) 7 eigenvectors

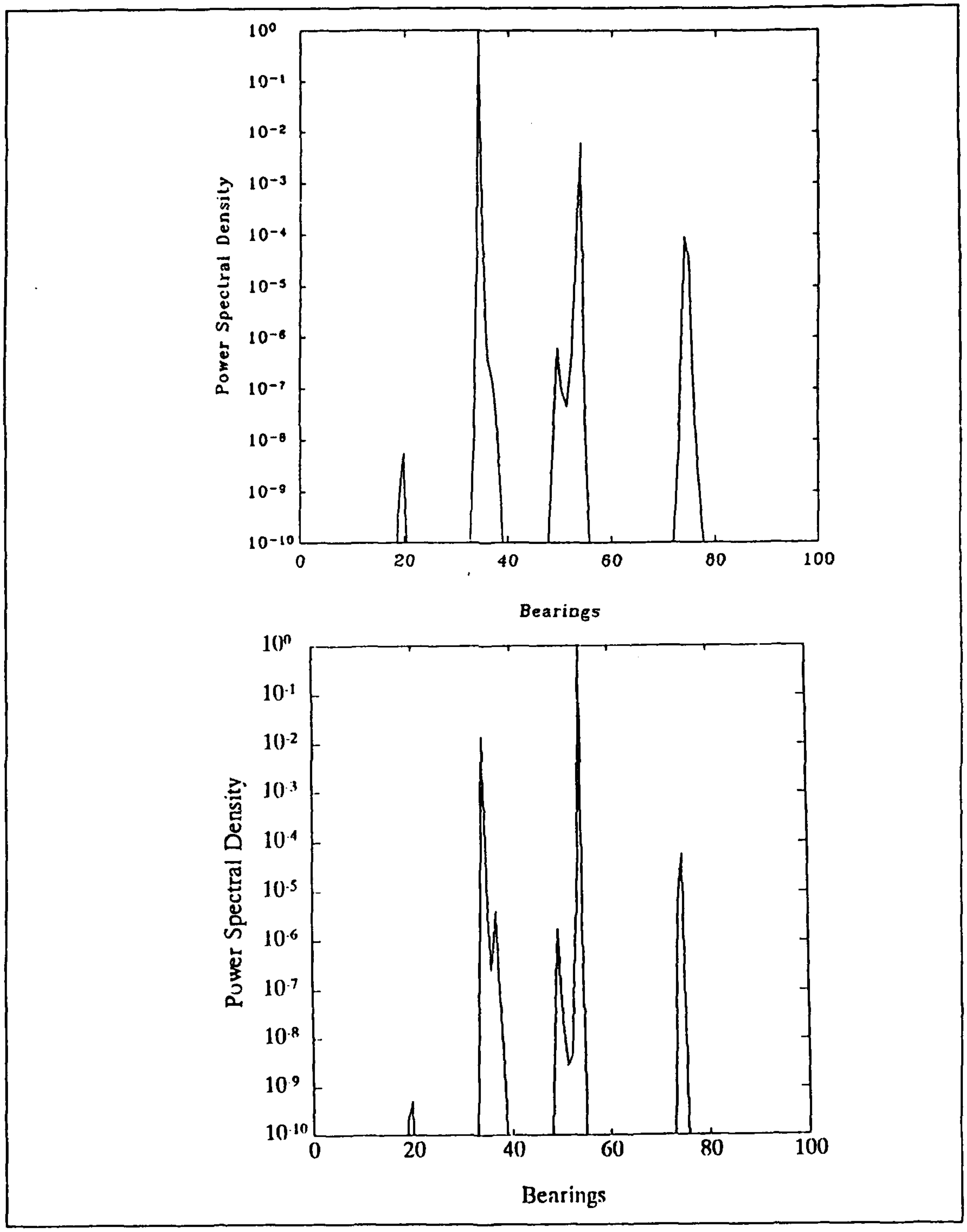

Figure 17. **Three targets at 34°, 36° and 54° (–10 dB) :**

product of the PSDs (a) 5 eigenvectors (b) 7 eigenvectors

D. COMPARISON

In this section we compare the performance of the block Lanczos algorithm and the single vector Lanczos algorithm for DOA estimation. In both cases, we chose the seven smallest eigenvalues and corresponding eigenvectors and use the spectral multiplication method since this gave the best preformance for both algorithms.

Example 1 is the comparison for detection of two targets at 18° and 45° for different SNRs. Figure 18(a) shows the result using the single vector Lanczos algorithm and Figure 18(b) indicates the result using the block Lanczos algorithm at an SNR of 0 dB. The result of the single vector case has two small spurious peaks; the block case has one very small spurious peak. In both cases the true bearings are clearly distinguished from the spurious peaks. Figure 19 shows a comparison of the results at –5 dB. The performance of the block Lanczos algorithm (Figure 19(b)) is much better than the single vector case (Figure 19(a)) even though several large spurious peaks are present. Figure 20(a) is the result at –10 dB using the single vector Lanczos algorithm and Figure 20(b) is the corresponding result of the block case. In the single vector case, the spurious peaks are almost the same as the true peaks and we cannot distinguish the true bearing location. In the block case, however, the true peaks are slightly larger than the largest spurious peak.

Example 2 is a comparison for detection of four targets at 18°, 27°, 29°, and 45°. Figure 21 shows the performance of the two methods at 0 dB. The resolution is clearly achieved in both cases. Figure 22 shows the results at –5 dB. Figure 22(a) shows that the single vector Lanczos algorithm cannot separate the two closly spaced targets. The block case (Figure 22(b)) shows slightly better resolution between targets located at 27° and 29°. At –10 dB (Figure 23(a) and Figure 23(b)) neither method can separate the two closely spaced in bearings at 27° and 29°.

Also, a spurious peak is produced by both methods.

The block Lanczos algorithm is known to estimate the multiple eigenvalues better than the single vector case [Ref. 1]. This situation is applicable to the DOA estimation using noise subspace computation where the noise is white. As observed in the results of Figures 18 – 23, the spectral estimation performance of the block Lanczos algorithm is consistently better than the single vector algorithm, particulary at low SNRs. Also, the block method provided better spectral resolution than the single vector method in our tests. While more analysis is needed to validate the simulation results, the overall performance of the block method is quite encouraging.

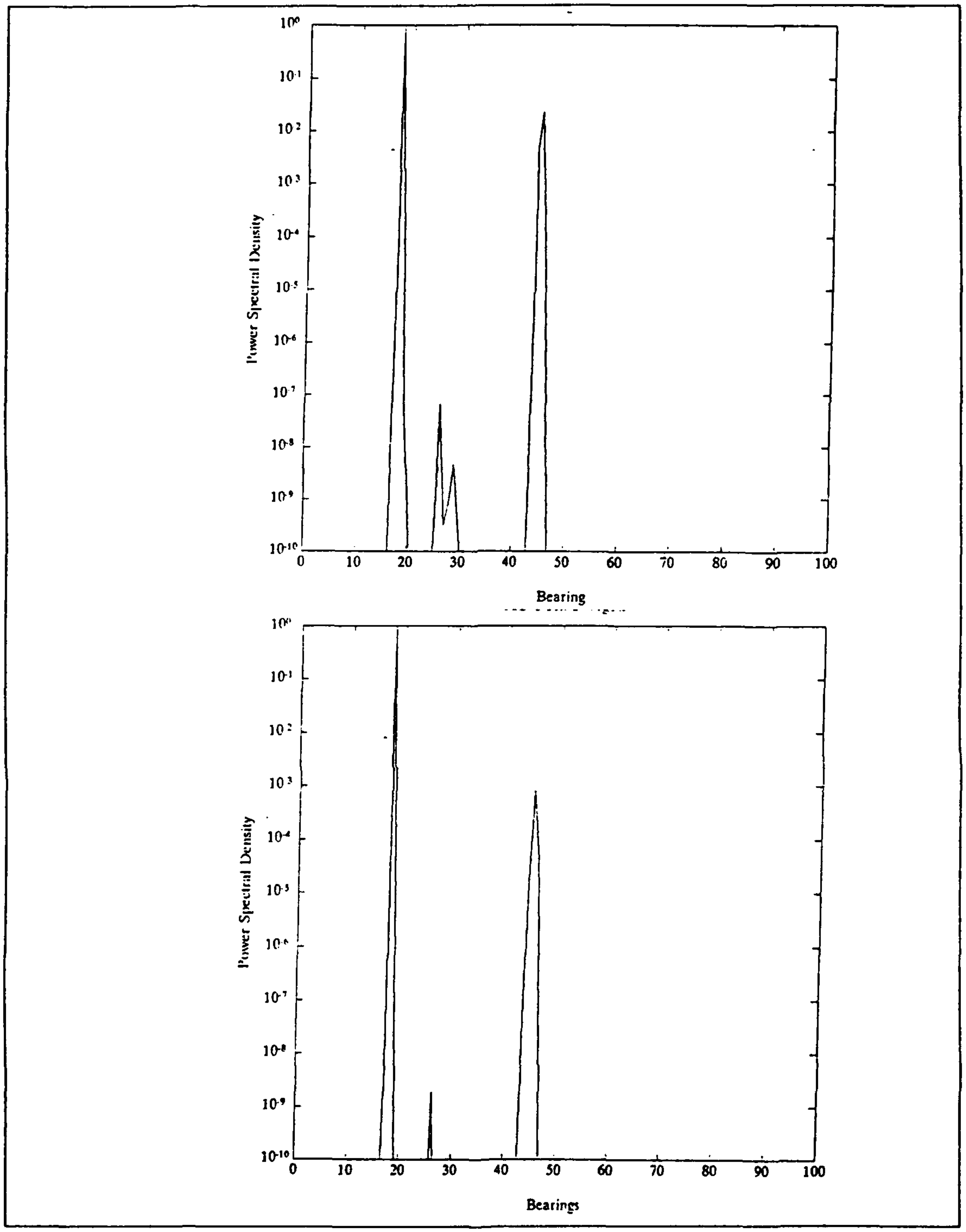

Figure 18. Two targets at 18° and 45° (0 dB) :

(a) single vector Lanczos method (b) block Lanczos method

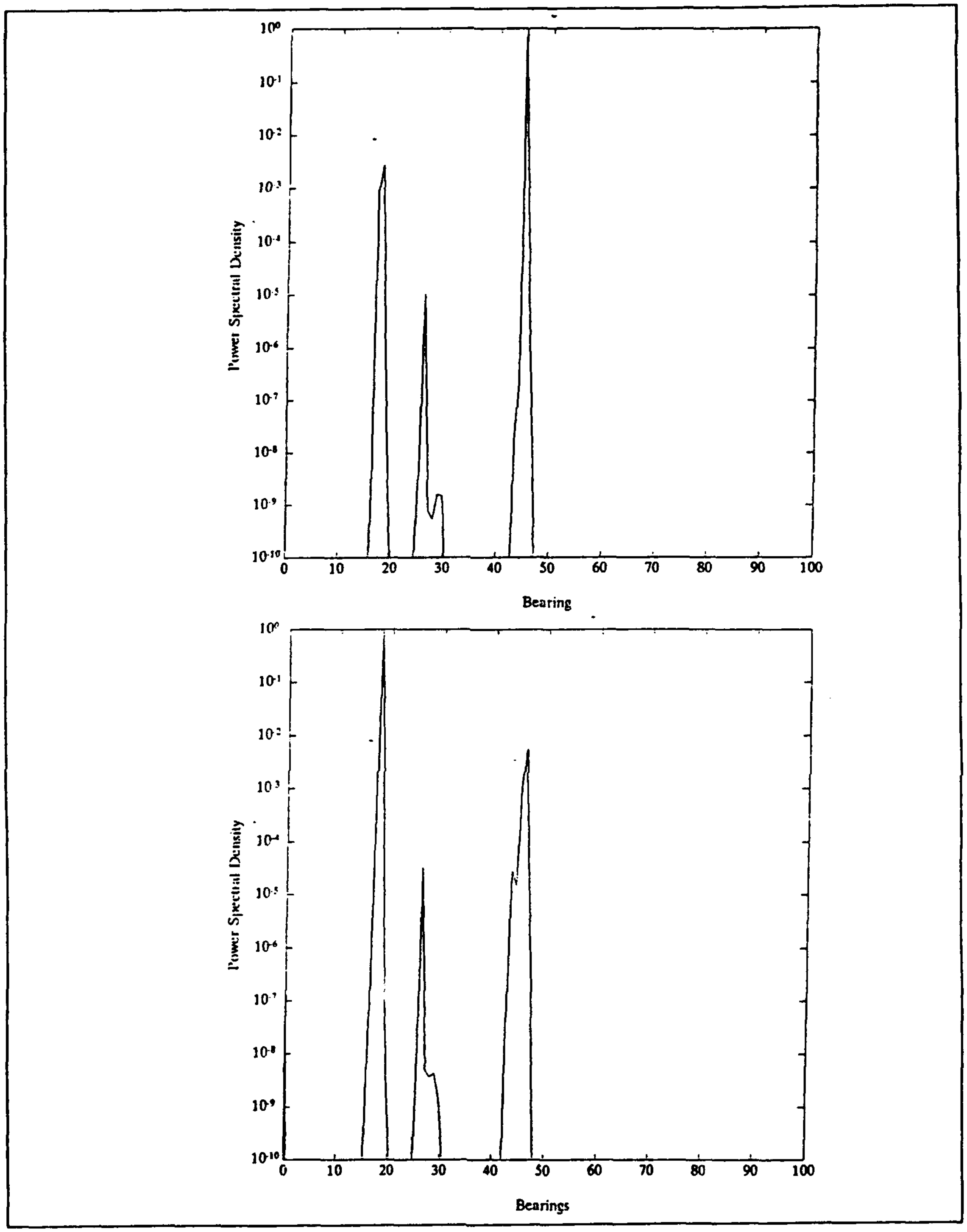

Figure 19. Two targets at 18° and 45° (–5 dB) :

(a) single vector Lanczos method (b) block Lanczos method

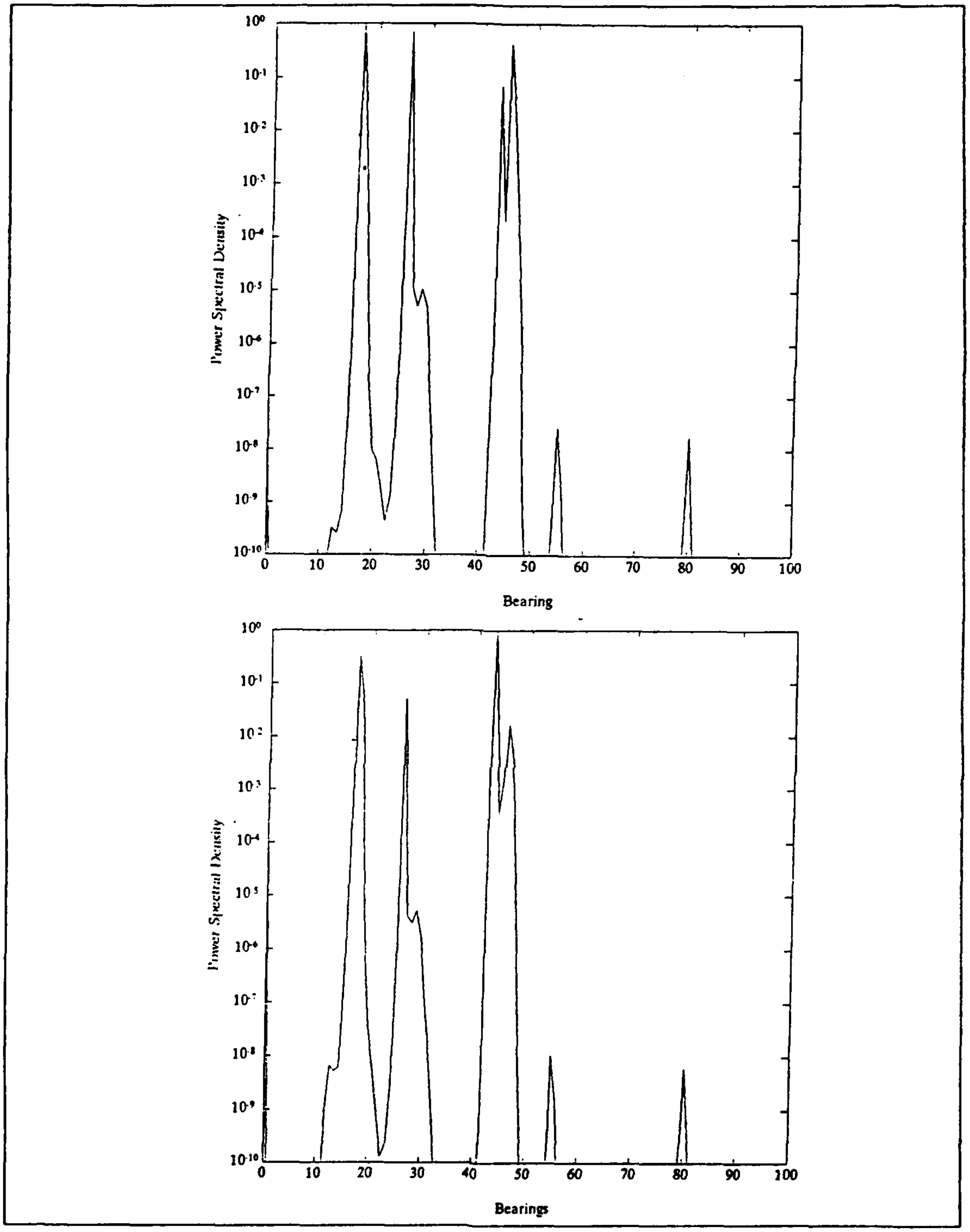

Figure 20. Two targets at 18° and 45° (–10 dB) :

(a) single vector Lanczos method (b) block Lanczos method

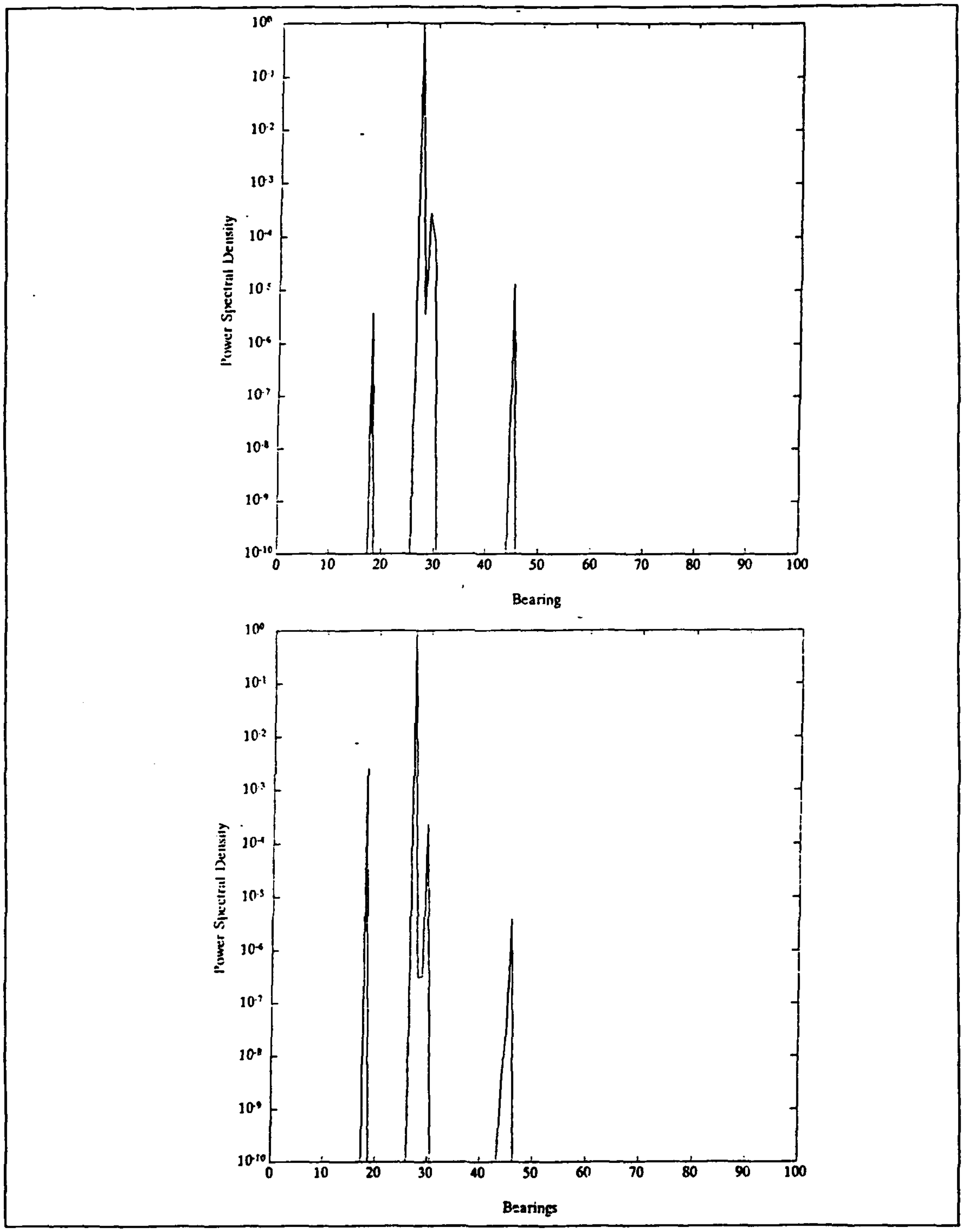

Figure 21. Four targets at 18°, 27°, 29° and 45° (0 dB) :

(a) single vector Lanczos method (b) block Lanczos method

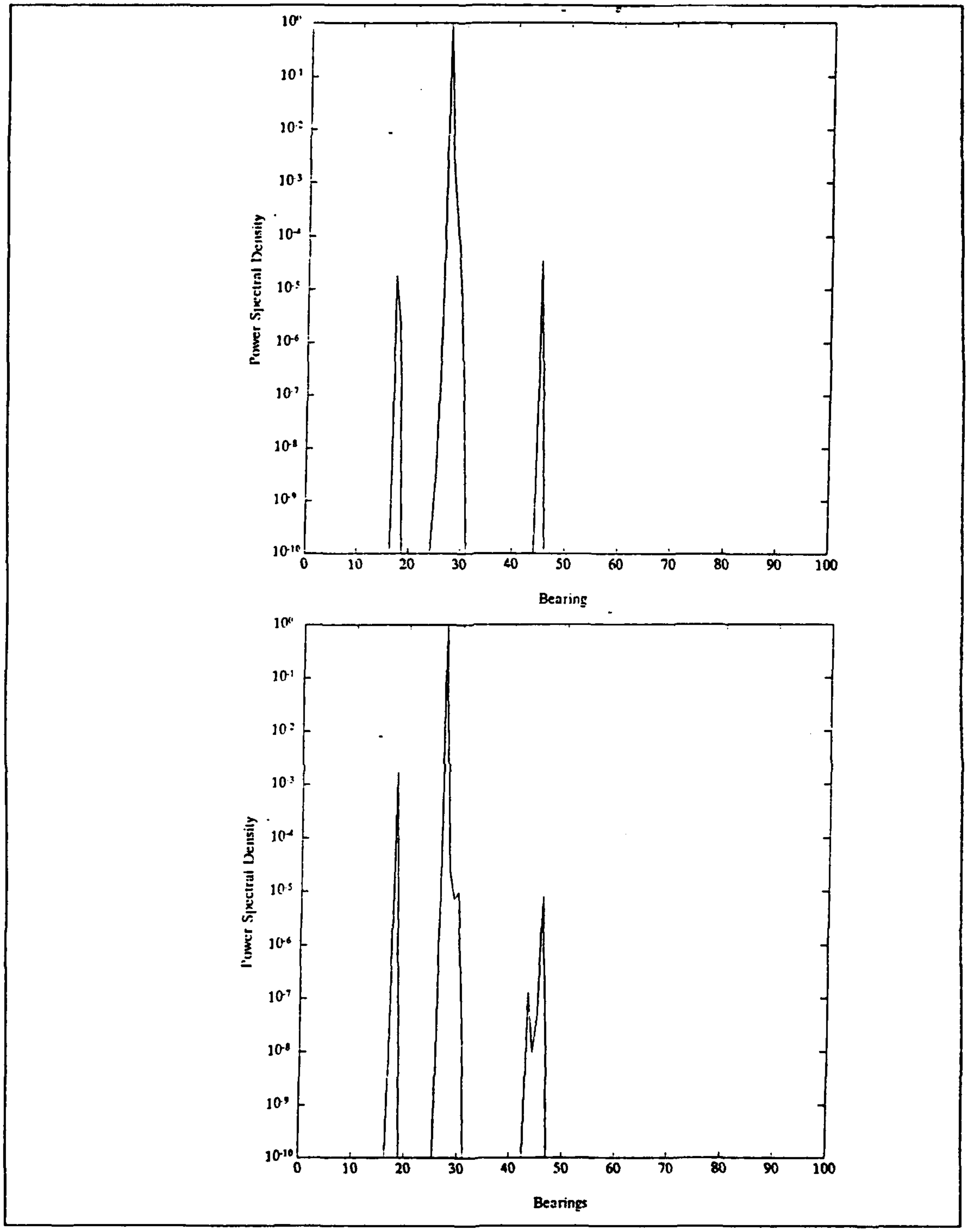

Figure 22. **Four targets at 18°, 27°, 29° and 45° (–5 dB) :**

(a) single vector Lanczos method (b) block Lanczos method

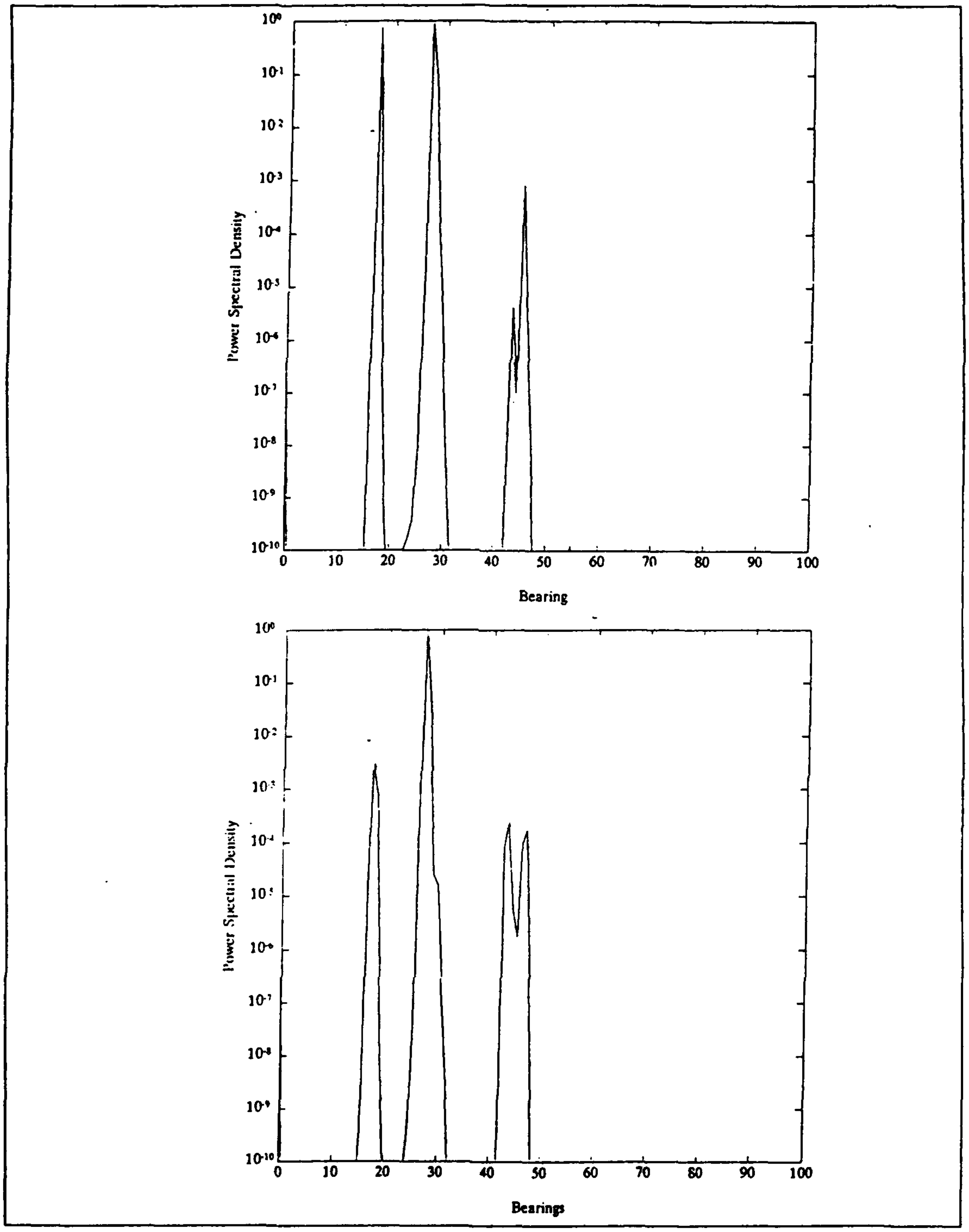

Figure 23. Four targets at 18°, 27°, 29° and 45° (–10 dB) :

(a) single vector Lanczos method (b) block Lanczos method

IV. SUMMARY AND CONCLUSIONS

In this thesis we examined the use of the single vector Lanczos method and the block Lanczos method and its application to spectral analysis and direction-of-arrival problems.

We computed a few of the extreme eigenvalues and their associated eigenvectors of a large symmetric matrix using the block Lanczos method. The eigenvalues and eigenvectors of the Lanczos matrix T_s approximate the corresponding eigenvalues and eigenvectors of the given matrix R. The block Lanczos algorithm can directly determine the multiplicities of the effective eigenvalues and the eigenvectors of R. We found that the spectral estimate of the block Lanczos method is more accurate than the single vector Lanczos method, particulary at low SNRs. Since we compute only a few of the extreme eigenpairs of a large autocorrelation matrix, the result of this algorithm is savings in computations and storage. This algorithm may be applied to any system where one needs to obtain a rapid decomposition of a large correlation matrix.

Although the results of this thesis are most encouraging, some additional work still remains to be done. We need to compare the results in computational speed and accuracy with other eigendecomposition techniques for valdating this algorithm. Also, we need to analyze the effect of roundoff errors on the eigenpair estimation of the Lanczos algorithm.

LIST OF REFERENCES

1. J. K. Cullum, and R. A. Willoughby (1985), *Lanczos algorithms for large symmetric eigenvalue computations, Vol. I.* IBM T. J. Watson Research Center.

2. E. Davidson (1975), "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of a large real–symmetric matrix", *J. Comput. Phys., 17,* pp. 87–94.

3. D. R. Fuhrmann (1988), "An algorithm for subspace computation, with applications in signal processing", *SIAM J. Matrix Anal. Appl.* vol. 9, No. 2. pp. 213–220.

4. G. H. Golub, and R. Underwood (1977), "The block Lanczos method for computing eigenvalues", In *Mathematical Software III,* J. R. Rice(Ed.), Academic Press, New York, pp. 361–377.

5. G. H. Golub, and C. F. Van Loan (1983), *Matrix Computation,* Johns Hopkins Press, Baltimore.

6. W. Kahan, and B. N. Parlett (1976), "How far should you go with the Lanczos process?", *Sparse Matrix Computations*, J. Bunch, and D. Rose, Academic Press, New York, pp. 131–144.

7. M. D. Kent (1989), *Chebyshev, Krylov, Lanczos: Matrix Relationships and Computations,* Ph. D. Thesis, Stanford University.

8. C. Lanczos (1956), *Applied Analysis,* Prentice–Hall, Englewood Cliffs, NJ.

9. C. Lanczos (1950), "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators", *Journal of Research of the National Bureau of Standards,* Vol. 45, No. 4, pp. 255–282.

10. C. C. Paige (1980), "Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem", *Linear Algebra and its Applications,* 34, American Elsevier Publishing, pp. 235–258.

11. C. C. Paige (1976), "Error Analysis of the Lanczos Algorithm for Tridiagonalizing a Symmetric Matrix", *J. Inst. Maths Applics,* 18, pp. 341–349.

12. C. C. Paige (1972), "Computational Variants of the Lanczos Method for the Eigenproblem", *J. Inst. Maths Applics,* 10, pp. 373–381.

13 C. C. Paige (1972), *An Error Analysis of a Method for Solving Matrix Equation*, Report STAN–CS–72–297, Computer Science Department, Stanford University.

14. B. N. Parlett (1980), *The Symmetirc Eigenvalue Problem,* Prentice–Hall, Englewood Cliffs.

15. B. N. Parlett, and D. S. Scott (1979), "The Lanczos algorithm with selective reorthogonalization", *Math. Comp. 33,* pp. 217–238.

16. J. H. Wilkinson (1965), *The Algebraic Eigenvalue Problem,* Oxford Science Publication.

17. P. Comon, and G. H. Golub (1989), *Tracking a few extreme singular values and vectors in signal processing*, Computer Science Department, Stanford University.

18. J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart (1976), "Reorthogonalization and Stable Algorithm for Updating the Gram–Schmidt QR Factorization", *Mathematics of Computation*, Vol. 30, No. 136, pp.772–795.

19. D. R. Fuhrmann, and B. Lie (1984), "An Iterative Algorithm for Locating the Minimal Eigenvector of a Symmetric Matrix", *ICASSP 84*, pp. 44.8.1–4.

20. D. W. Tufts, and C. D. Melissinos (1986), "Simple, Effective Computation of Principal Eigenvectors and their Eigenvalues and Application to High–Resolution Estimation of Frequencies", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP–34, pp. 1046–1052.

21. L. V. Kantorovich, and G. P. Akilov (1982), *Functional Analysis*, Pergamon Press, Oxford, England.

22. D. E. Gear (1989), *DOA estimation by eigendecomposition using single vector Lanczos algorithm*, Master's Thesis, Naval Postgraduate School, Monterey, California.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor Murali Tummala, Code 62Tu Department of Electrical and Computer Engineering Monterey, CA 93943-5000	3
5.	Professor C. W. Therrien, Code 62Ti Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Professor Ralph Hippenstiel, Code 62Hi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
7.	Dr. Rabi Madan Office of Naval Research Code 1114 800 North Quincy Street Arlington, VA 22217-5000	1
8.	Library, P.O.Box 77 GongNeung-Dong, DoBong-Ku, Seoul Republic of Korea, 130-09	1
9.	Yong Joo Kim 164-57 Namgajoa 1-Dong SeoDaeMun-Ku, Seoul Republic of Korea, 121	5